Ansys 2023 R1

Maxwell Help

ANSYS, Inc.
Southpointe
2600 Ansys Drive
Canonsburg, PA 15317
ansysinfo@ansys.com
https://www.ansys.com
(T) 724-746-3304
(F) 724-514-9494

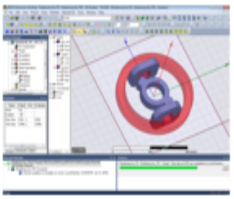
Release 2023 R1
January 2023

Table of Contents

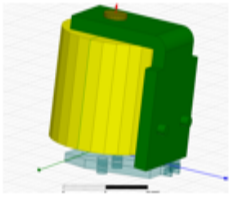
1 - Welcome to Maxwell Help

Use the following links for quick information on the following topics.

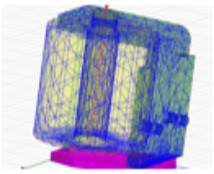
Getting Started Maxwell



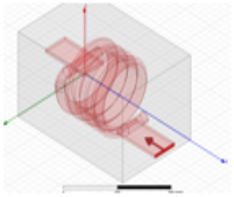
Maxwell User Interface



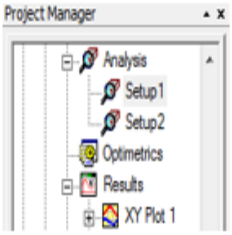
Modeling



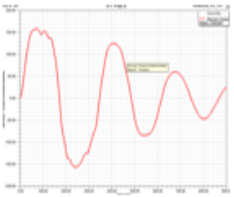
Meshing



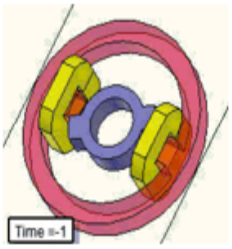
Boundaries and Excitations



Analysis



Results



Example Projects

Optimetrics

Scripting

Use the following links for Maxwell Process Flow topics:

User Interface Quick Links

Use the following links for quick information on the following topics.

The Maxwell Desktop	Working with Ribbons
General Purpose Keyboard Shortcuts for Maxwell	Running Maxwell From a Command Line

For detailed information on these and many other topics:

- Use **F1** on any open dialog to open the Help for that dialog.
- With the cursor, hover over any menu command, icon, or window and press **F1** for help on that selection.
- With the Help **Contents tab** selected, navigate the help topic hierarchy.
- With the Help **Index tab** selected, search the help index.
- In the Help **Search** box, search the full help text.

Modeling Quick Links

Use the following links for quick information on the following topics.

Set the model's units of measurement.	Setting Modeler Drawing Options
Assign transparency to an object.	Selecting Items in the Modeler Window
Subtract objects.	Drawing Objects
Measuring Objects	Choosing the Cursor Movement Mode
Drawing Bondwires	Importing Files
Keyboard shortcuts for the 3D Modeler Window.	Modifying Objects
Modifying the Model View	

The help system provides different ways to find information and navigate quickly:

- Press **F1** on any open dialog to open the Help for that dialog.
- A *hierarchical table of contents* – You can browse through the table of contents, expand entries, and close entries. Click on an entry to see it in the content area.
- A *full text search* – To locate every occurrence of a word or phrase that may be contained in the help, use the Search Function.

Boundaries/Excitations Quick Links

Use the following links for quick information on the following topics.

Defining Boundary Conditions	Defining Excitations
Setting Eddy Effects and Displacement Current	Setting Core Loss
Setting Up Motion for Transient Projects	Functional Boundaries and Excitations
Defining Boundary Conditions in 2D	Defining 2D Excitations

The help system provides different ways to find information and navigate quickly:

- Press **F1** on any open dialog to open the Help for that dialog.
- A *hierarchical table of contents* – You can browse through the table of contents, expand entries, and close entries. Click on an entry to see it in the content area.
- A *full text search* – To locate every occurrence of a word or phrase that may be contained in the help, use the search function.

Meshing Quick Links

Use the following links for quick information on the following topics.

Defining Mesh Operations	Plot the finite element mesh
Detecting and Addressing Model Problems to Improve Meshing	Handling Complicated Models

The help system provides different ways to find information and navigate quickly:

- Press **F1** on any open dialog to open the Help for that dialog.
- A *hierarchical table of contents* – You can browse through the table of contents, expand entries, and close entries. Click on an entry to see it in the content area.
- A *full text search* – To locate every occurrence of a word or phrase that may be contained in the help, use the search function.

Analysis Quick Links

Use the following links for quick information on the following topics.

Specifying the Analysis Options	Remote Analysis
Configuring Distributed Analysis	Specifying Solution Settings
Selecting an Optimal Configuration for Distributed Analysis	Setting Adaptive Analysis Parameters

The help system provides different ways to find information and navigate quickly:

- Press **F1** on any open dialog to open the Help for that dialog.
- A *hierarchical table of contents* – You can browse through the table of contents, expand entries, and close entries. Click on an entry to see it in the content area.
- A *full text search* – To locate every occurrence of a word or phrase that may be contained in the help, use the search function.

Optimetrics Quick Links

Use the following links for quick information on the following topics.

Setting up a Parametric Analysis	Setting up an Optimization Analysis
Setting up a Sensitivity Analysis	Tuning a Variable
Setting up a Statistical Analysis	Setting a Range function
Setup Calculations for Optimetrics	Adding a cost function

The help system provides different ways to find information and navigate quickly:

- Press **F1** on any open dialog to open the Help for that dialog.
- A *hierarchical table of contents* – You can browse through the table of contents, expand entries, and close entries. Click on an entry to see it in the content area.
- A *full text search* – To locate every occurrence of a word or phrase that may be contained in the help, use the search function.

Results Quick Links

Use the following links for quick information on the following topics.

View solution data	Creating Reports
Plot field overlay	Working with Traces
Plot the finite element mesh	Adding Data Markers to Traces
Create animations	Creating a Quick Report
Specifying Output Variables	Dynamically Updating Reports During Solution
Using the Fields Calculator	

The help system provides different ways to find information and navigate quickly:

- Press **F1** on any open dialog to open the Help for that dialog.
- A *hierarchical table of contents* – You can browse through the table of contents, expand entries, and close entries. Click on an entry to see it in the content area.
- A *full text search* – To locate every occurrence of a word or phrase that may be contained in the help, use the search function.

Scripting Quick Links

Use the following links for quick information on the following topics.

Recording a Script	Running a script
Stopping Script Recording	Pausing and Resuming a Script

Stopping a Script	

The help system provides different ways to find information and navigate quickly:

- Press **F1** on any open dialog to open the Help for that dialog.
- A *hierarchical table of contents* – You can browse through the table of contents, expand entries, and close entries. Click on an entry to see it in the content area.
- A *full text search* – To locate every occurrence of a word or phrase that may be contained in the help, use the search function.

Maxwell Getting Started Guides

Maxwell documentation includes the following *Getting Started Guides*:

- *3D Rotational Actuator with Parametrics*

  This Getting Started Guide is written for Maxwell beginners and experienced users who would like to quickly familiarize themselves with the capabilities of Maxwell 3D. This guide leads you step-by-step through creating, solving, and analyzing the results of solving a 3D rotational actuator magnetostatic problem.

  By following the steps in this guide, you will learn how to perform the following tasks:
  - Draw a geometric model.
  - Modify a model's design parameters.
  - Assign variables to a model's design parameters.
  - Specify solution settings for a design.
  - Validate a design's setup.
  - Run a Maxwell simulation.
  - Plot the magnetic flux density vector.
  - Run a parametric analysis.
  - Create an animation using saved parametric field data.
- *3D Transient Rotational Actuator Problem with Motion*

  This Getting Started Guide is written for Maxwell beginners and experienced users who would like to quickly familiarize themselves with the capabilities of Maxwell 3D. This guide leads you step-by-step through creating, solving, and analyzing the results of solving a Transient rotational actuator problem with motion. This Getting Started Guide builds on the problem and techniques presented in the 3D rotational actuator magnetostatic problem.

  By following the steps in this guide, you will learn how to perform the following tasks:
  - Import a geometric model.
  - Modify a model's design parameters.
  - Assign variables to a model's design parameters.
  - Specify solution settings for a design.
  - Validate a design's setup.
  - Run a Maxwell simulation.
  - Plot the magnetic flux density vector.
  - Include motion in the simulation.

- *2D Magnetostatic Solenoid Problem with Parametrics*

  This Getting Started Guide is written for Maxwell beginners and experienced users who would like to quickly familiarize themselves with the capabilities of Maxwell 2D. This guide leads you step-by-step through creating, solving, and analyzing the results of solving a 2D asymmetric solenoid magnetostatic problem with parametric analysis.

  By following the steps in this guide, you will learn how to perform the following tasks:
  - Draw a geometric model.
  - Modify a model's design parameters.
  - Assign variables to a model's design parameters.
  - Specify solution settings for a design.
  - Validate a design's setup.
  - Run a Maxwell 2D simulation.
  - Plot the simulation results.
  - Set up a parametric sweep analysis varying multiple solution variables.
  - Plot families of curves of the parametric results.
- *Skin Depth Seeding*

  By following the steps in this guide, you will learn to use Skin Depth-Based mesh refinement for HFSS or Maxwell Modelers to specify a number of layers and a skin depth as an alternative to modeling physical layers. The guide also shows how to use mesh plots with cut planes, as well as a repurposed **Model Analysis** dialog box to view and obtain statistics on layering success.

2 - Getting Started with Maxwell and RMxprt

Maxwell is an interactive software package that uses finite element analysis (FEA) to solve three-dimensional (3D) electrostatic, magnetostatic, eddy current, and transient problems.

Using Maxwell you can compute:

- Static electric fields, forces, torques, and capacitances caused by voltage distributions and charges.
- Static magnetic fields, forces, torques, and inductances caused by DC currents, static external magnetic fields, and permanent magnets.
- Time-varying magnetic fields, forces, torques, and impedances caused by AC currents and oscillating external magnetic fields.
- Transient magnetic fields caused by electrical sources and permanent magnets.

RMxprt™ is an interactive software package used for designing and analyzing electrical machines.

Maxwell and Rmxprt are integrated into the Ansys Electronics Desktop.

Ansys Electronics Desktop Student

Ansys Electronics Desktop Student is a free Windows version of the Electronics Desktop that allows you to model, mesh, solve, and post-process in HFSS, Q3D Extractor, 2D Extractor, Maxwell 3D and Maxwell 2D, RMxprt, and Icepak. The following list of limitations applies to all supported design types in Ansys Electronics Desktop Student.

> **Note:** Unsupported design types (e.g., Mechanical) do not open in Ansys Electronics Desktop Student.

- Geometry export not supported
- Import of DXF and STEP files only
- Local solve only (remote configuration not supported)
- High Performance Computing limited to 4 cores
- Integration with Ansys Workbench not supported
- Beta features not supported

HFSS Student Limitations

- SBR+ not supported
- Mesh assemblies not supported
- Mesh element count limit (analysis and post-processing)
    - 3D volume: 64,000 elements
    - 3D surface: 8,000 elements
    - 2D: 2,000 triangles

Q3D Extractor and 2D Extractor Student Limitations

- Mesh element count limit (analysis and post-processing)
  - 3D volume: 64,000 elements
  - 3D surface: 8,000 elements
  - 2D: 2,000 triangles

Maxwell and RMxprt Student Limitations

- Mesh element count limit (analysis and post-processing)
  - 3D volume: 64,000 elements
  - 3D surface: 8,000 elements
  - 2D: 2,000 triangles
- 3D and 2D Transient simulations

Icepak Student Limitations

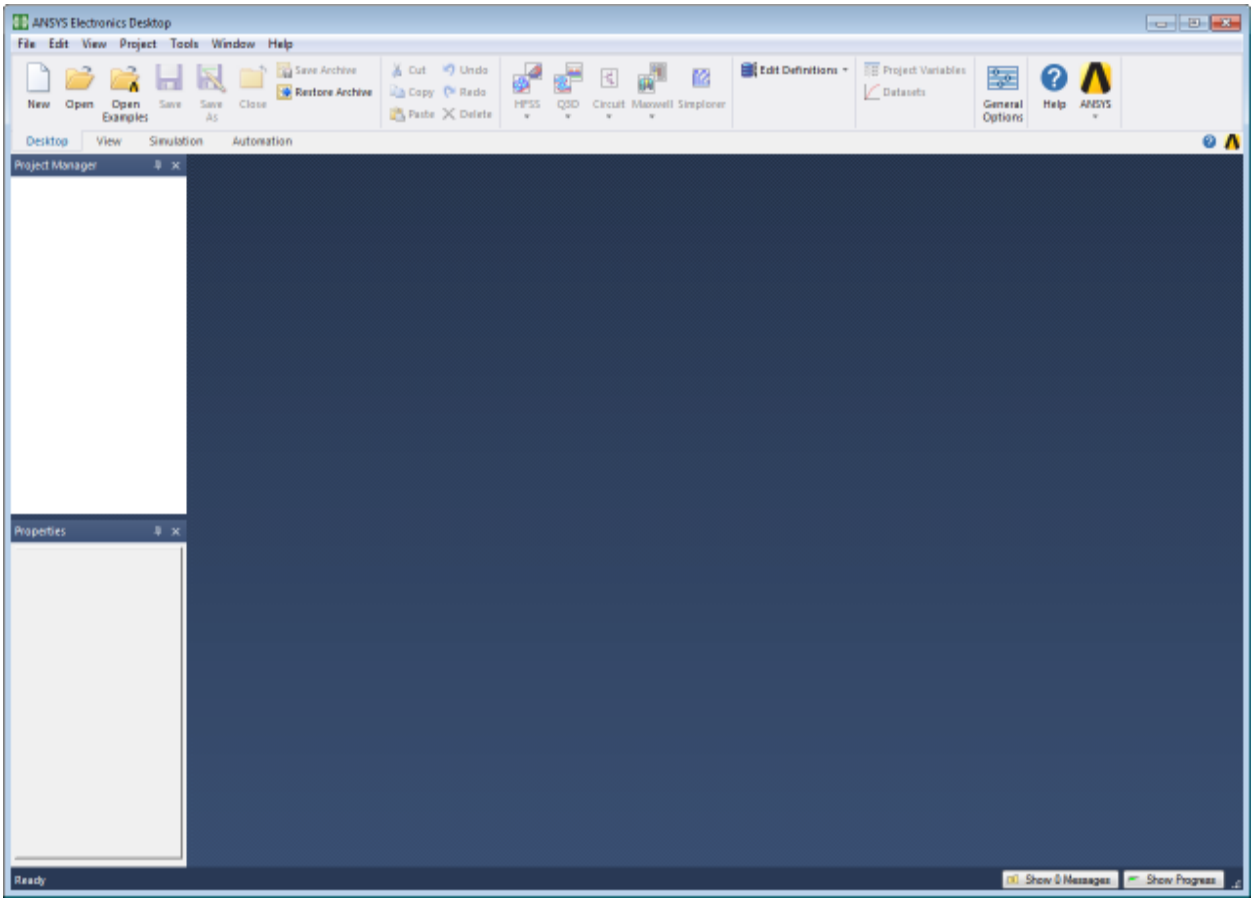
- Mesh element count limit (analysis and post-processing): 512,000 elements

Related Topics

Working with Maxwell Projects and Designs
Setting Up a Maxwell Design
Getting Started with RMxprt
Quick Start for RMxprt
Using the Maxwell Circuit Editor

Welcome to Ansys Electronics Desktop

The Ansys Electronics Desktop, illustrated in the following figure, provides a comprehensive environment for designing and simulating various electronic components and devices. The Ansys Electronics Desktop consists of a unified user interface where a wide variety of design types can be created. Typically, you can create or import a design, set up the simulation, validate your design, run the analysis, and post process the results.



The desktop has the following design types and features:

- HFSS - a general purpose 3D interface for the design, analysis, and simulation of electromagnetic components.
- HFSS 3D Layout - a full-wave layout-based electromagnetic simulator with a specialized interface for geometries created in layout.
- HFSS-IE - a full wave Integral Equation solver for large open problems.
- Q3D Extractor - a quasi-static 3D solver for extracting lumped RLGC parameters and Spice models.
- 2D Extractor - a 2D solver for extracting per-unit-length RLGC parameters of transmission lines.
- Circuit - a schematic-based interface to the Nexxim circuit simulator.
- Circuit Netlist - a netlist (text-based) interface to the Nexxim circuit simulator.
- Maxwell 3D - uses finite element analysis (FEA) to solve three-dimensional (3D) electrostatic, magnetostatic, eddy current, and transient problems.
- Maxwell 2D - uses finite element analysis (FEA) to solve two-dimensional (2D) electrostatic, magnetostatic, eddy current, and transient problems.

- RMxprt - a template-based electrical machine design tool that provides fast, analytical calculations of machine performance and 2-D and 3-D geometry creation for detailed finite element calculations in Ansys® Maxwell®.
- Maxwell Circuit Design - sets up external circuit designs to supply excitations to coil terminals for Maxwell 2D and 3D Eddy Current and Transient designs.
- Twin Builder - an integrated, multi-domain, mixed-signal simulator for complex technical systems.
- Icepak - a general purpose 3D interface for the design, simulation, and thermal analysis of electronic components
- Launch Savant - a tool for simulating near-field, far-field, and coupling performance of antennas installed on platforms and within other complex and electrically large environments.
- Launch EMIT - a simulation tool for the prediction of radio frequency interference (RFI) in complex environments.

If you go to the **Project** menu, you can access all of the design types. Any combination of these design types can be inserted into a single project file. The schematics can be used to wire up the different field solver models and create a model of a high-level system. The Ansys Electronics Desktop provides an efficient way to manage complicated projects that require several different analysis tools to model all of its pieces. Designs can also be parameterized. With the help of the Optimetrics feature the best design variations can be made available to other modules when the designs are linked into a higher level simulation. This lets you study the effect of varying a design parameter on the behavior of the entire system.

Related Topics

"Launching Ansys Electronics Desktop" below

Launching Ansys Electronics Desktop

After you have installed Ansys Electronics Desktop, start the program by one of the following methods:

Maxwell contains the following menus, which appear at the top of the desktop:

- Double-click the **Ansys Electronics Desktop** icon on your desktop.
- Click **Ansys Electronics Desktop** on the Windows Start menu.
- Click **Start>All Programs>Ansys EM Suite 2023 R1>Ansys Electronics Desktop 2023 R1** on the Windows Start menu.

Related Topics

"Welcome to Ansys Electronics Desktop " on page 2-2

Obtaining Information about the Software and Release

To obtain information about the software:

1. Click **Help>About Ansys Electronics Desktop** to open the **About Ansys® Electromagnetics Suite [release number]** dialog.
2. Click the **Installed Components** tab to view a list of software installed.
3. Click the **Client License Settings** tab to view information about the following:
   - Provider Name
   - ANSYSLI Version
   - FlexNet Publisher Servers
   - Admin Directory
   - Customer Number
   - FLEXlm Version
   - Redirect Info
4. To export the software information:
   a. Click **Export**.
   The **Save As** dialog box appears.
   b. Browse to the location where you want to save the information as a text file.
   c. Type a name for the file in the **File name** text box. The **Save as type** pull-down list is already specified as **Export (*.txt)**.
   d. Click **Save**.
5. Click **OK** to close the **About Ansys® Electromagnetics Suite [release number]** dialog.

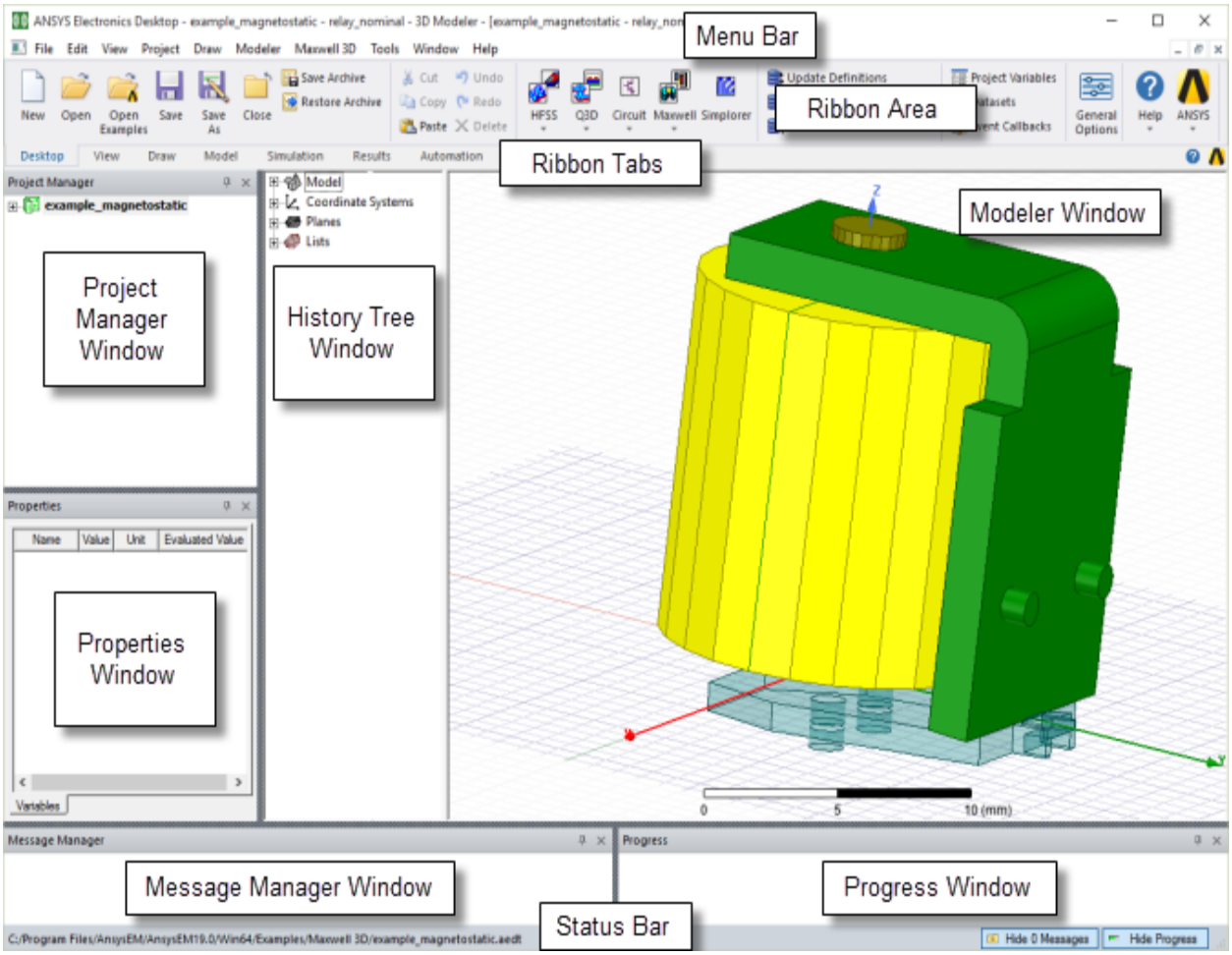
System Requirements

Maxwell supports certain versions of Windows and Linux. For supported platforms and system requirements, go to:

http://www.ansys.com/Solutions/Solutions-by-Role/IT-Professionals/Platform-Support

The Maxwell Desktop UI

The Maxwell desktop consists of several windows, a menu bar, ribbons, and a status bar. You can open multiple windows in Maxwell to display different parts of the model. For instance, one window can remain fixed on the winding, one on the diagram, and one on the main desktop window. To open a new window, click **Window>New Window**. To move back and forth between windows, select the **Windows** menu, and select the window you want to view.



Click a link below to view more information about that desktop component.

Menu Bar	Message Manager
Ribbon Area/Ribbon Tabs	Progress Window
Project Manager Window	Modeler Window
Properties Window	History Tree Window
Status Bar	

Working with the Menu Bar

The menu bar enables you to perform all Maxwell tasks, such as managing project files, customizing the desktop, drawing objects, and setting and modifying all project parameters.

Maxwell contains the following menus, which appear at the top of the desktop:

File menu	Use the **File** menu commands to manage Maxwell project files and printing options.
Edit menu	Use the **Edit** menu commands to modify the objects in the active model and undo and redo actions.
View menu	Use the **View** menu commands to display or hide desktop components and model objects, modify **2D** and **3D Modeler** window visual settings, and modify the model view. The **ACT Extensions**command opens the ACT Extensions window which lets you work with ACT integration tools.
Project menu	Use the **Project** menu commands to insert a Maxwell or RMxprt design (as well as other Desktop designs) to the active project, define datasets, and define project variables.
Draw menu	Use the **Draw** menu to draw the elements that make up a model.
Modeler menu	Use the **Modeler** menu commands to import, export, and copy 2D Modeler files and 3D Modeler files; assign materials to objects; manage the **3D Modeler** window's grid settings; define a list of objects or faces of objects; control surface settings; perform Boolean operations on objects; and set the units for the active design.
Maxwell3D, Maxwell2D, or RMxprt menu	Use these menu commands to change solution type, assign boundaries and excitations, set up parameters, add analysis setups, set up Optimetrics, post process solutions, export equivalent circuits, define datasets, and other design tasks.
Tools menu	Use the **Tools** menu to modify the active project's material library, arrange the material libraries, run and record scripts, update project definitions from libraries, display options, and modify many of the software's default settings.
Window menu	Use the Window menu commands to rearrange the **Modeler** windows.
Help menu	Use the **Help** menu commands to access the help system and view the current Maxwell version information.

Working with Windows

The **Window** menu allows you to manage the currently open windows in the **Maxwell** Desktop user environment.

In addition to the standard functions listed in the table, the menu also contains a list of currently open windows for easily switching between them. This is important in multiple window environments where the desktop make become crowded with open windows and some may be partially or completely obscured. Minimized windows are automatically opened when selected in this list.

Command	Description

New Window	Open a new window. New windows shows the model by default.
Cascade	Arranges all non-minimized windows in a cascade with the window title bar visible.
Tile Horizontally	Arranges all non-minimized windows to use the maximum horizontal size of the desktop.
Tile Vertically	Arranges all non-minimized windows to use the maximum vertical size of the desktop.
Arrange Icons	Arranges the icons of any minimized windows along the bottom of the Desktop.
Close All	Close All currently open desktop windows.

Moving and Resizing Desktop Windows

You can customize the appearance of the desktop by moving and resizing the Status Bar, the Message Manager, the Project Manager, the docked Properties window, and the Progress window.

To move one of these windows:

1. Click and hold on the title bar.
2. Drag the cursor toward the region where you want to place the window.

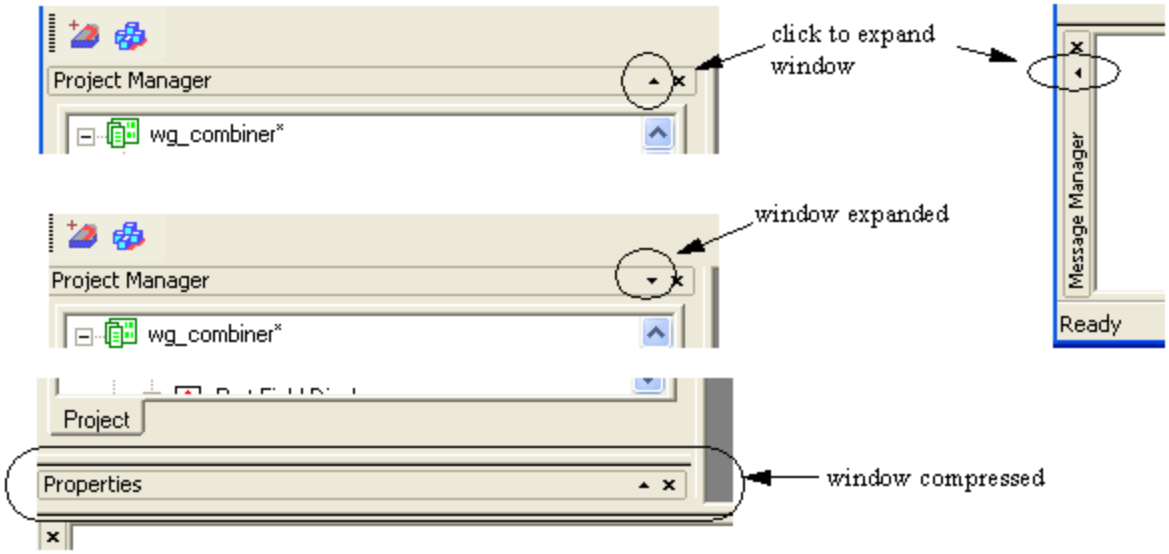
   A rectangle shape follows the cursor. As you drag the rectangle to different parts of the desktop, the changes in dimension show when you have reached a location where you can place the window. This can be at the top, left, bottom, and side of the modeler window. You can place a window next to another, as well as above or below another.

   If you drag the window to the center of the 3D Modeler window, you can place it there as a floating window.
3. Release the mouse button to place the window.

You can also resize the windows in two ways.

- To size a desktop window, place the cursor over an edge of the window. Over the inner-edges, for sizing a window within the desktop, the cursor changes to a double bar with arrows pointing each direction. Over the outer-edges, for sizing the desktop, the cursor changes to a line with arrows pointing each direction. Press and drag to size the window.
- To expand a window to fill the horizontal or vertical space it shares with another window, click the triangle in the window title bar. When you expand a window, the triangle appears as inverted and any other windows in the same horizontal or vertical space are compressed to only the title bar. If a window does not share a horizontal or vertical space with another,

the triangle does not appear.



Window Layouts

You can create and save specific window layouts for rhe desktop and apply them when needed. For example, you can create a layout that shows a large component window. There are options to apply a saved layout, restore the default window layout, and remove saved layouts.

To save a wndow layout:

1. On the **View** menu, click **Window Layouts>Save Current Layout**.
2. Enter the layout name in the **Layout Name** dialog box, and click **OK**. The layout is saved and added to the list of layouts on the **Window Layouts** submenu.

To apply a saved layout:

1. On the **View** menu, click **Window Layouts** and select from the listed saved layouts.

To restore the default window layout:

1. On the **View** menu, click **Window Layouts>Default**.

To remove window layouts:

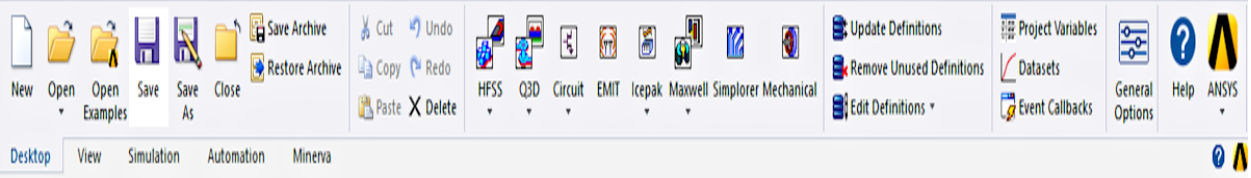
1. On the **View** menu, click **Window Layouts>Remove Saved Layouts**.
2. In the **Modify Layouts** dialog box, select one or more layouts and click **Delete**, or click **Delete All**.

Working with Ribbons

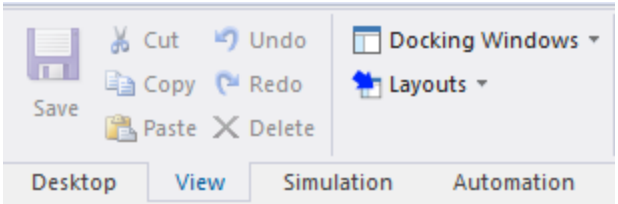
The ribbon is the rectangular area across the top of the application. It comprises various tabs, each one representing a subset of commands available from the menus. The initial set of tabs (Desktop, View, Simulation, Automation) offers commands for adding and opening projects,

selecting solvers, configuring the Desktop display (window choice, size, and position), configuring the simulation environment, setting scripted Event Callbacks or General Options, and showing Automation features for recording and using scripts. The **Desktop, View, Simulation, Automation,** and **Minerva** tabs appear for all design types.

Desktop Tab Selected



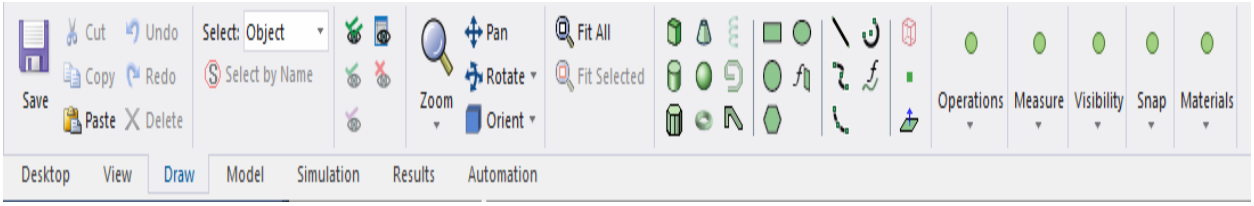
View Tab Selected without a Project



The Docking Windows and Layouts areas contain drop-down menus with additional selections.

If you have inserted a Project, for example, an HFSS project, the **View** ribbon displays commands appropriate for the active editor. For instance, the View tab will show different commands for the modeler (shown below) and the report viewer.

If you compress the Desktop window, each tab display shows fewer available features than an expanded display, with priority given to the most used features.

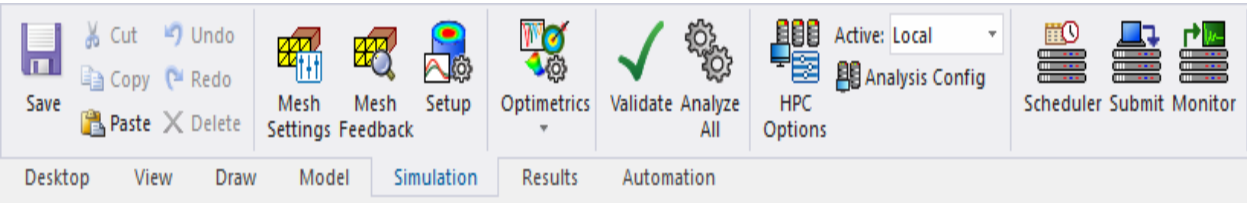


Draw Tab Selected

Model Tab Selected

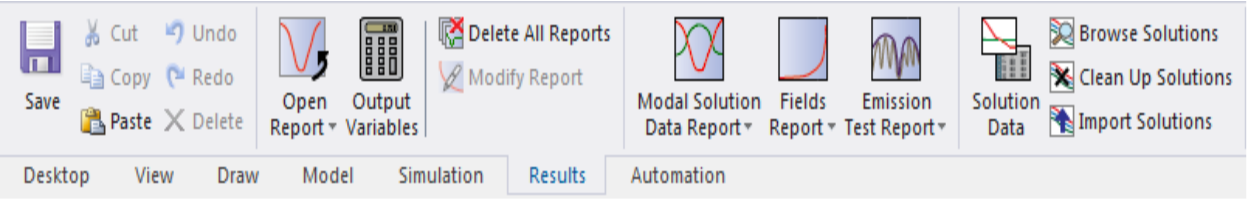
Simulation Tab Selected

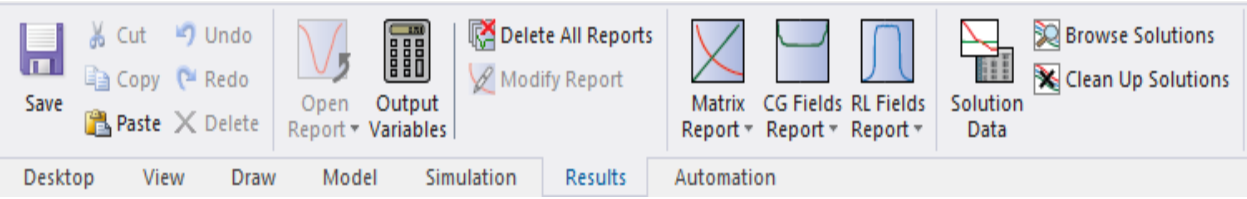
In this example, the **Simulation** tab contains commands appropriate for the currently active design.



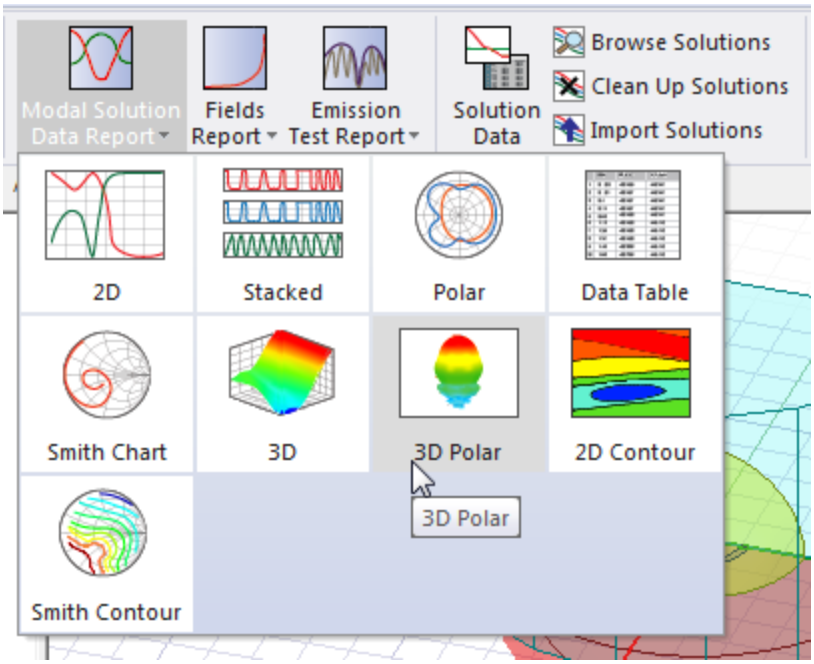
Results Tab Selected

In these examples, each **Results** tab contains commands appropriate for the currently active design.



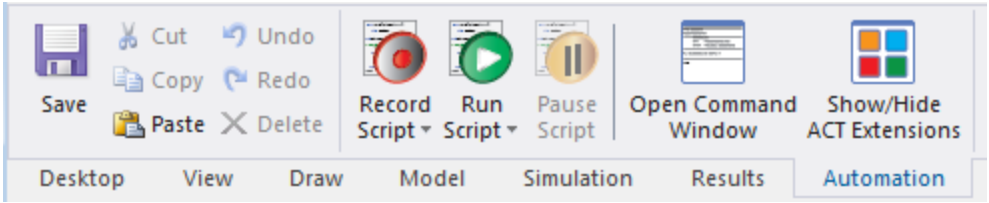


The drop-down menus under the Report icons let you select from available report types:



Automation Tab Selected

This tab includes controls for recording and running scripts, as well as access to the Command window, and display of ACT Extensions.



Related Topics

The Maxwell Desktop

External User Tools

To add an external user tools menu to Maxwell:

1. Click **Tools>External Tools**
2. This displays the **Customize User Tools Menu** dialog.

   If a **User Tools** menu has been defined, its contents are displayed. Navigation buttons let you Move Up, Move Down, Add, and Delete.
3. Click the **Add** button in the **Customize User Tools Menu** dialog.

This enables the following fields:

Menu Text field -- this displays [new tool] as text you will replace with the text you want to appear in the User Tools menu.

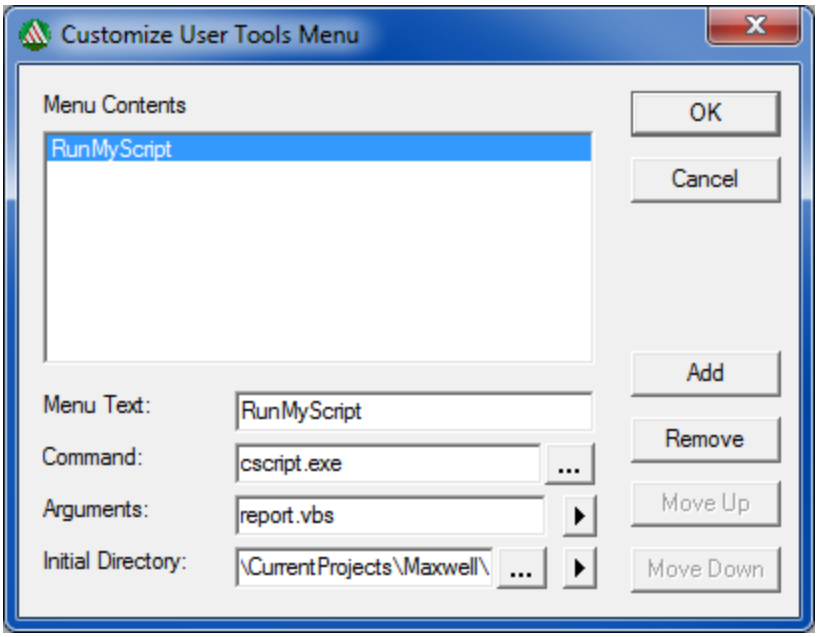
Command field -- this will display the external executable. An ellipsis button [...] lets you navigate to the file location.

Arguments field -- this field accepts command arguments from the **>** button menu selections for File Path, File Directory, File Name, File Extension, Project Directory, or Temp Directory.

Initial Directory – this field specifies the initial directory for the command to operate. The ellipsis button {...] displays a dialog that lets you navigate folders in your desktop, or across the network.

4. Click **OK** to add the External Tools menu to Maxwell or **Cancel** to close the dialog without changes.

You can also add scripts to the **Tools** menu. Assuming you have a script to generate custom reports called report.vbs, use the cscript.exe program to execute your script.



This example shows the cscript.exe program added to the **Tools** menu as **RunMyScript**. The command line argument to the csscript.exe program is report.vbs. You can also name the directory in which it will be run.

Working with the Shortcut Menus

A variety of shortcut menus — menus that appear when you right-click a selection — are available in the ribbons area of the desktop, in the **Modeler** window, and in the **Project Manager** window.

Shortcut menu in the Modeler window	Use the shortcut menu in the **Modeler** window to select, magnify, and move options (zoom, rotate, etc.), change the view, perform boolean operations, assign materials, or mesh operations to objects, and work with field overlays.
Shortcut menus in the Project Manager window	Use the shortcut menus in the **Project Manager** window to manage Maxwell project and design files and design properties; assign and edit excitations, and mesh operations; add, analyze, and manage solution setups; add optimetrics analyses; create post-processing reports; edit project definitions.
Shortcut menus in the History Tree	Use the shortcut menus in the History tree to expand or collapse groupings. If you select particular objects in the history tree, the shortcut menu lists the commands that you can apply to the selected object or objects.

Note	All of the commands on the shortcut menus are also available on the menu bar.

Shortcut Menu in the Modeler Window

Use the shortcut menu in the **Modeler** window to select, magnify, and move options (zoom, rotate, etc.), change the view, assign materials, nets, excitations, or mesh operations to objects, work with field overlays, and plot objects.

To access the shortcut menu in the **Modeler** window:

- Right-click in the **Modeler** window (grid area).

Shortcut Menus in the Project Manager Window

Each node, or item, in the project tree has a shortcut menu.

To access the shortcut menu in the **Project Manager** window, for a particular node:

- Select a node or item.
- Right-click in the **Project Manager** window.

General Purpose Keyboard Shortcuts for Maxwell

The following keyboard shortcuts apply to Maxwell and RMxprt in general

- F1: Help
- F1 + Shift: Context help
- F4 + CTRL: Close program
- CTRL + C: Copy
- CTRL + N: New project
- CTRL + O: Open...

- CTRL + P: Print...
- CTRL + V: Paste
- CTRL + X: Cut
- CTRL + Y: Redo
- CTRL + Z: Undo
- CTRL + 0: Cascade windows
- CTRL + 1: Tile windows horizontally
- CTRL + 2: Tile windows vertically

To customize the shortcut assignments, use Tools>Keyboard Shortcuts.

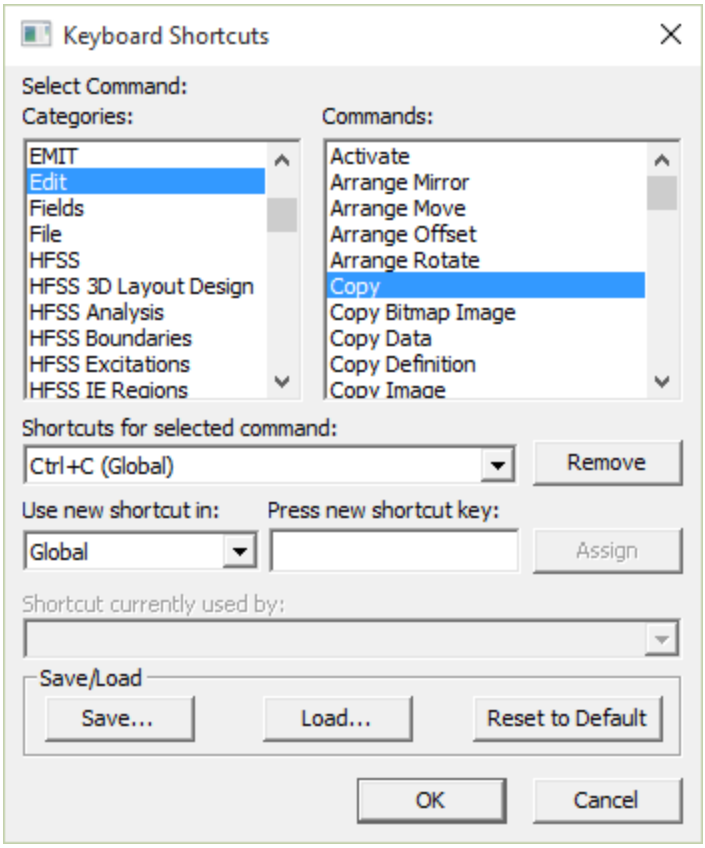
Related Topics

Keyboard Shortcuts for the 3D Modeler Window

Custom Keyboard Shortcuts

Custom Keyboard Shortcuts

Click **Tools>Keyboard Shortcuts** to display a dialog that lets you view existing assignments, create new shortcuts, save and load assignment files.



Selecting a Command category lists the available Commands for that category. If the command has an assigned shortcut, it is displayed in the **Shortcuts for selected command** field. You can use the **Remove** button to disable the shortcut for the selected command. If the selected command does not have an assigned shortcut, the **Shortcuts for selected command** field and the **Remove** button are grayed out.

To create a new shortcut key:

1. Select the Category and Command.
2. If you want disable a current assignment for the selected command, click **Remove**.
3. To assign a keyboard shortcut, place the cursor in the **Press new shortcut key** field.

   The field displays the keystrokes you make. When you have made keystrokes, the dialog enables the **Assign** button. If you combine keystrokes these are displayed with a "+" between them. For example, Ctrl + p or Alt +o.
4. The **Use new shortcut in** field displays **Global** by default, which means that the shortcut will apply to all applicable contexts. If a limited context exists, the menu will offer a selection.
5. When you have made the desired assignments, you can save the assignments to a named file.

   Clicking the **Save...** button displays a browser window that lets you navigate the file structure and assign a name, using an **aks** suffix for Ansoft Keyboard Shortcut file.

   Buttons on the browser window let you designate the file location as **Use Path**, **Personallib**, **Syslib**, **UserLib**, or **Project** folder.

   If you have an existing **aks** file, you can use the **Load...** button to display a browser window to locate the desired file.
6. You can **OK** the current settings, or **Reset to Default**.

Working with the Status Bar

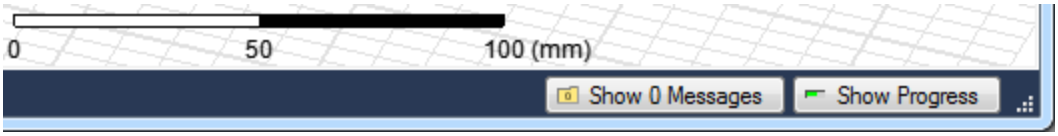
The status bar is located at the bottom of the application window. It displays information about the command currently being performed.

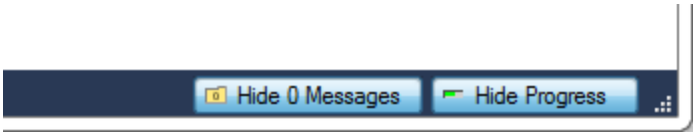
To display or hide the status bar:

- Click **View>StatusBar**.

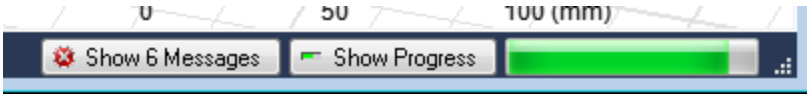
  A check box appears next to this command if the status bar is visible.

  When visible, the status bar contains buttons to show or hide the message window and progress window.





When more than one progress bar is active, the top progress bar is represented on the status bar with a progress indicator.



Depending on the command being performed, the status bar can display the following:

- **X**, **Y**, and **Z** coordinate boxes.
- A pull-down list to enter a point's absolute, relative, cartesian, cylindrical, or spherical coordinates.
- The model's units of measurement.

When creating object in the model window, you may jump to the coordinate entry boxes by pressing the **Tab** key. In addition, the **Tab** key moves the focus to the next coordinate box when performing manual data entry.

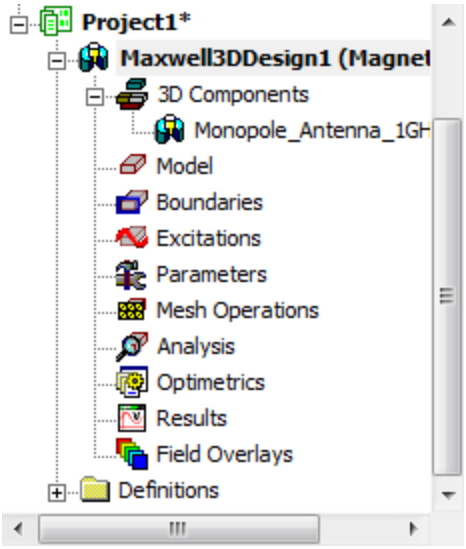
Exiting the Desktop

To exit the desktop, use **File>Exit**.

If simulations are running, a message informs you that if you continue, the simulations will be aborted. If you choose **OK**, Maxwell does a clean abort before going on with the Exit.

Working with the Project Manager

The **Project Manager** window displays the open project's structure, which is referred to as the project tree.



The **Project Manager** window displays details about all open Maxwell projects. Each project ultimately includes a geometric model, its boundary conditions and excitations, and field solution and post-processing information. Any 3D Components appear under the 3D Components icon.

To show or hide the **Project Manager**:

- Click **View>Project Manager**.
  A check box appears next to this command if the **Project Manager** window is visible.

Related Topics

Working with the Project Tree

Shortcut Menus in the Project Manager Window

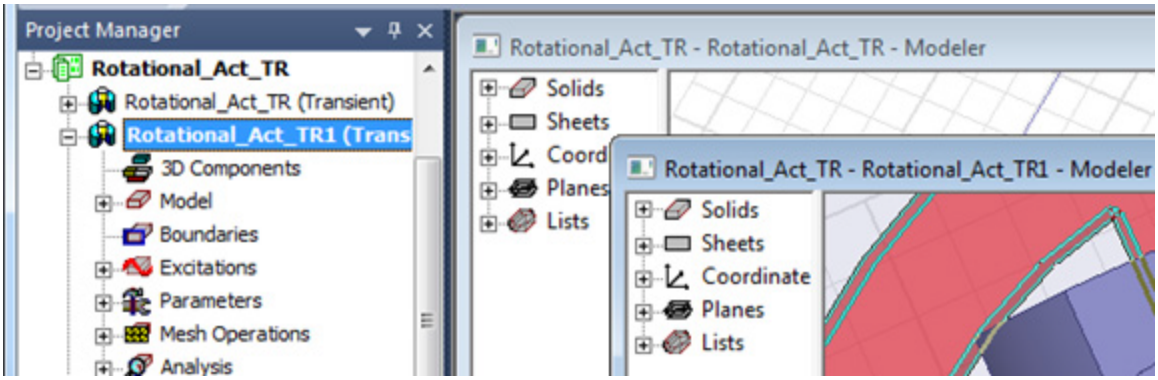
"Working with the RMxprt Project Manager" on page 26-12

Working with the Project Tree

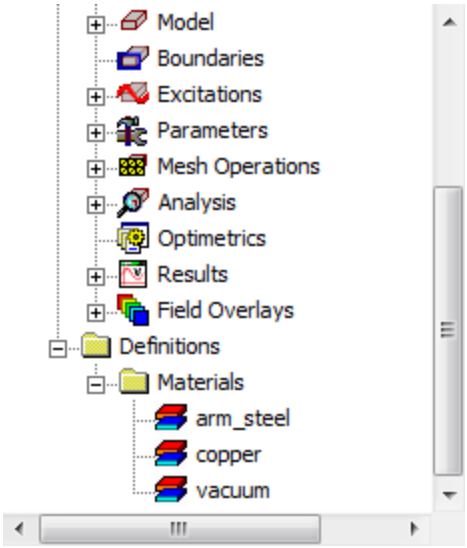
The project tree is located in the **Project Manager** window and contains details about all open Maxwell projects.

The top node listed in the project tree is the project name. It is named **Projectn** by default, where *n* is the order in which the project was added to the current session of Maxwell. Expand the project icon to view the project's Maxwell design information, 3D Components (if any) and material definitions.

By default, the Project tree icon for the active window is highlighted, as shown in the following figure. (See User Interface for the options.) With the option disabled, **Rotational_Act_TR (Transient)** in the Project tree would not be highlighted.



When the Project tree is taller than the window, if you mouse has a scroll wheel, you can click in the Project manager and use the wheel to scroll up and down. If the Project tree is larger than the window size, there is also a scroll bar can control with the mouse cursor.



In the project tree, an "unknown" section lists objects that do not fit one of the following dimensionality types: solid, sheet, or wire.

Related Topics

Viewing Maxwell Design Details

Setting the Project Tree to Expand Automatically

You can set the project tree to automatically expand when an item is added to a project.

1. Click **Tools>Options>General Options**.

   The **Options** dialog box appears.

2. Click the **Project Options** tab.
3. Under **Additional Options**, select **Expand Project Tree on Insert**.
4. Click **OK**.

Viewing Maxwell Design Details

Once you insert a Maxwell design into a project, it is listed as the second node in the project tree. It is named MaxwellModel*n* by default, where *n* is the order in which the design was added to the project. Expand the design icon in the project tree to view specific data about the model.

The MaxwellModel*n* node contains the following project details:

3D Components	Lists any 3D components added to the design.
Model	Allows you to display the design list, define motion setup, and set the symmetry multiplier. For Transient solutions only, when the Model icon is selected, the Properties window displays a Motion Variables tab showing Speed and Direction variables values (not user-editable) if a motion setup has been defined.
Boundaries	Allows you to define boundary conditions on selected faces.
Excitations	Allows you to define voltage and current sources on selected objects.
Parameters	Allows you to assign executive parameters to solve for (such as force and torque).
Mesh	Displays the mesh operations specified for objects or object faces. Mesh operations are optional mesh refinement settings that are specified before a mesh is generated.
Analysis	Displays the solution setups for a Maxwell design. A solution setup specifies how Maxwell computes the solution.
Optimetrics	Displays any Optimetrics setups added to a Maxwell design.
Results	Displays any post-processing reports that have been generated.
Field Overlays	Displays field overlay plots, which are representations of basic or derived field quantities on surfaces or objects. Plot folders are listed under **Field Overlays.** These folders store the project's plots and can be customized. See *Setting Field Plot Defaults* for information on how to customize the plot folders.

Note	To edit a project's design details: • In the project tree, double-click the design setup icon that you want to edit. A dialog box appears with that setup's parameters, which you can then edit.

In the project tree, an "unknown" section lists objects that are not solid, sheet, or wire.

Design List Dialog Box

You can view the design list for a model in the following ways:

- By clicking **Maxwell3D**, **Maxwell2D**, or **RMxprt** and then selecting **List**.
- By double-clicking **Model** in the project tree.
- By right-clicking items in the project tree and selecting **List** from the shortcut menu.

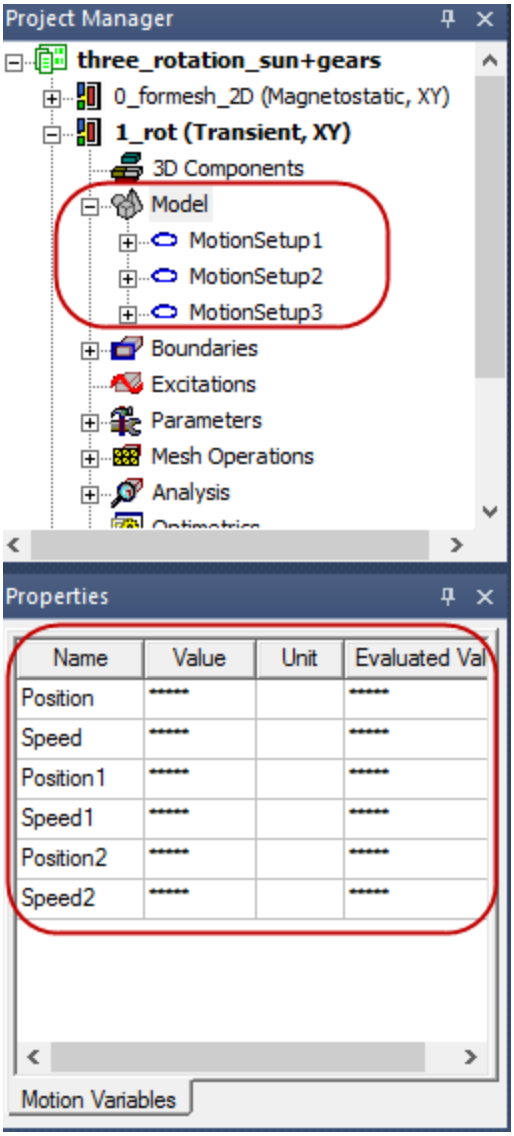
Information about the model, boundaries and excitations, mesh operations, and analysis setup can be viewed in the **Design List** dialog box that appears, with the following tabs available:

- **Model**
- **Machine** (for RMxprt projects)
- **Motion**
- **Boundaries**
- **Excitations**
- **Parameters**
- **Mesh Operations**
- **Analysis Setup**

In the **Design List** dialog box, all objects are shown, regardless of the type of dimensionality (solid, sheet, wire, or unknown); whereas, in the project tree, an "unknown" section lists objects that are not solid, sheet, or wire.

Motion Variables Tab

For 3DTransient solutions, when the Model icon in the project tree is selected, the **Properties** window displays a **Motion Variables** tab showing **Speed** and **Position** variables values if a motion setup has been defined. Similarly, for 2D transient solutions, which support multiple motion setups, the tab shows speed and position variables corresponding to each motion setup: **Position** and **Speed** for MotionSetup1; **Position1**, **Speed1** for MotionSetup2; **Position2**, **Speed2** for MotionSetup3, and so forth. Below is an example with three motion setups.



Note	Position and speed variables are treated as intrinsic variables and are not user-editable. Their values are dynamically calculated during solve time and are expressed in standard SI units. While these variables are *related* to motion setups, they are not necessarily *used* in motion setups. For example, they may also be used with the transient design's excitations. Like other variables, they can be used in expressions and reports.

Related Topics

Setting Up Motion for Transient Projects

Using Intrinsic Variables

Viewing Material Definitions

The definitions node is listed at the bottom of the project tree and displays all of the material definitions that are assigned to the objects in the active model.

Related Topics

Adding New Materials

Working with the Properties Window

The **Properties** window displays the attributes, or properties, of an item selected in the project tree, the history tree, or the **Modeler** window. The **Properties** window enables you to edit an item's properties. The properties, and the ability to edit them in the **Properties** window vary depending on the type of item selected. The tabs available in the **Properties** window also vary depending the selection.

You can choose to show or hide a docked **Properties** window as part of the desktop. You can move and resize the docked Properties window within the desktop to suit your work style. When you have a docked Properties window, it displays the properties of any item you select in the Project tree, the History Tree, or the Modeler window. You select **View>Properties Window** to toggle the viewing of the docked properties window.

Regardless of the whether you display a docked **Properties** window, you can still open an undocked **Properties** window for any item in the project tree, the history Tree, or the 3D Modeler window by double-clicking.

Related Topics

Opening the Properties Window
Showing and Hiding the Properties Window
Auto-Complete for Variables and Properties in Electronics Desktop
Setting the Properties Window to Open Automatically
Modifying Object Attributes using the Properties Window
Modifying Object Command Properties Using the Properties Window
Working with the RMxprt Properties Window

Opening the Properties Window

1. Select the object whose properties you want to view.
2. Click **Edit>Properties**.
   The **Properties** window for that object appears.
3. When you are finishing making changes, click **OK**.

Rather than opening a separate window, you can also view an object's properties if you have the **Properties** window displayed within the desktop.

Related Topics

Showing and Hiding the Properties Window
Setting the Properties Window to Open Automatically
Modifying Object Attributes using the Properties Window
Modifying Object Command Properties Using the Properties Window

Showing and Hiding the Properties Window

To show or hide the **Properties** window on the desktop:

- Click **View>Properties Window**.

A check box appears next to this command if the **Properties** window is visible.

Note	You can also click any command or attribute in the **Property** window, and then click **Edit>Properties** to view its properties.

Related Topics

Setting the Properties Window to Open Automatically

Setting the Properties Window to Open Automatically

To modify the object's properties and set the **Properties** window to open after an object is drawn, do the following:

1. Click **Tools>Options>Modeler Options**.
   The **Modeler Options** window appears.
2. Click the **Drawing** tab.
3. Select **Edit property of new primitives**.
4. Click **OK**.
   Hereafter, after you draw an object in point mode, the **Properties** window will open. However, if you draw an object in Dialog mode, this setting is ignored.

Related Topics

Showing and Hiding the Properties Window

Modifying Object Attributes Using the Properties Window

1. Select the object for which you want to edit its attributes by clicking it in the view window or clicking its name in the history tree.
2. Under the **Attribute** tab in the **Properties** window, edit the object attribute.

   Depending on the attribute type, you can edit it by doing one of the following:
   - Select the check box to apply the attribute; clear the check box to disable the attribute.
   - Click in the field and edit the numeric values or text, and then press **Enter**.
   - Click the button and then edit the current settings in the window or dialog box that appears.
   - Click the **Value** column of the attribute, and then select a new setting from the menu that appears.

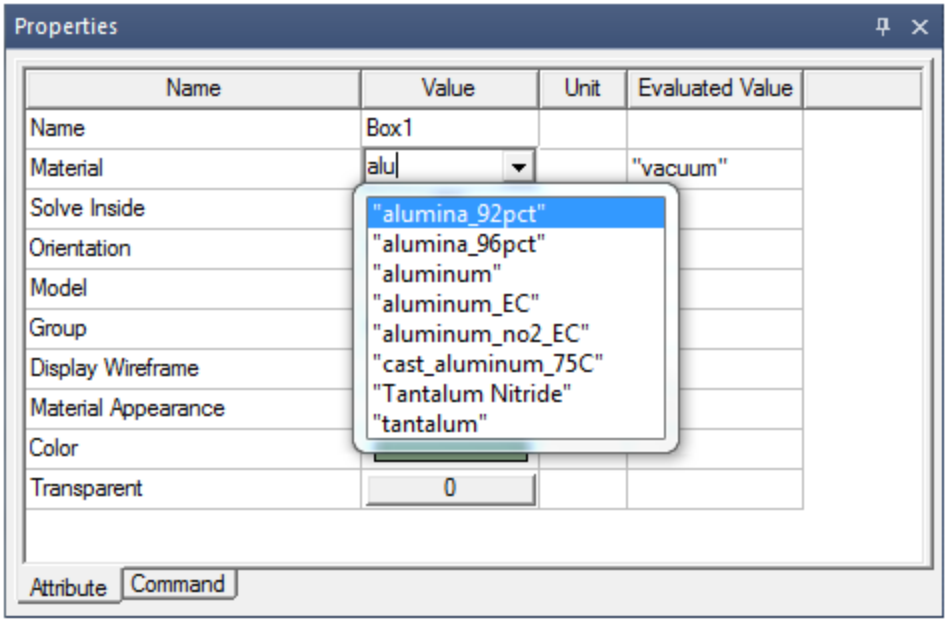
Related Topics

Modifying Objects

Modifying Object Command Properties Using the Properties Window

Auto-Complete for Variables and Properties in Electronics Desktop

When you edit a properties or variable text field, Ansys Electronics Desktop can display possible matches for what you type. This can help if, for example, a variable or material name is long. You can save time by selecting a predetermined match rather than typing out the entire name. The following figure shows an example of auto-complete for material names:



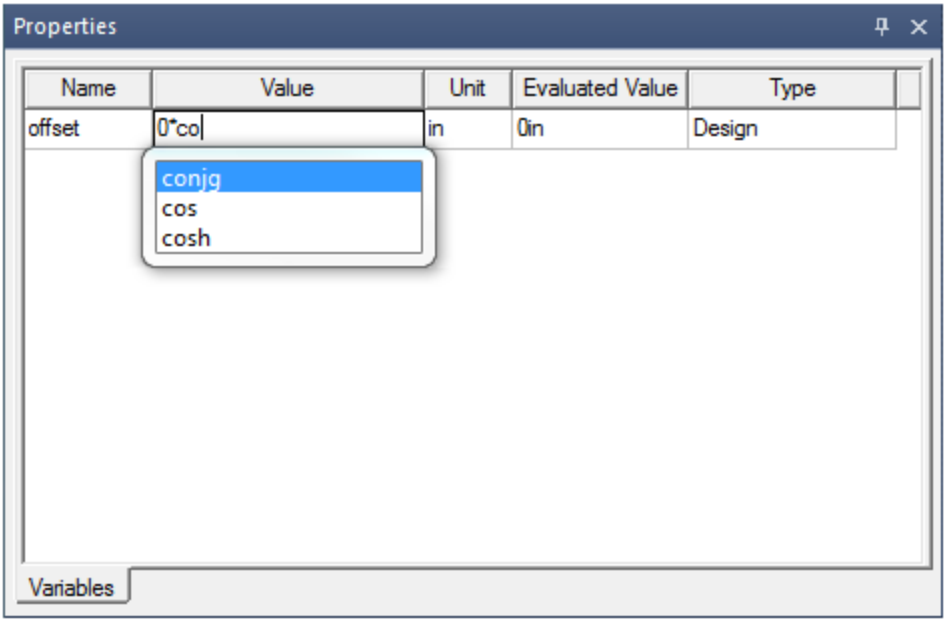
Using Auto-Complete

Certain commonly used text fields have auto-complete configured. (Not all editable fields are configured to work with auto-complete.) When you start typing in a field that is configured to work with auto-complete, items that match display in a list window below the text field. If there is no matching text, then no list window displays. With 10 matches or fewer, the list window is vertically sized according to the number of matches. The width of the window is based on the width of the text field. You can re-size the list window, and the new size is "remembered" when doing further auto-complete matching in the same text field. Switching to a different text field resets the sizing.

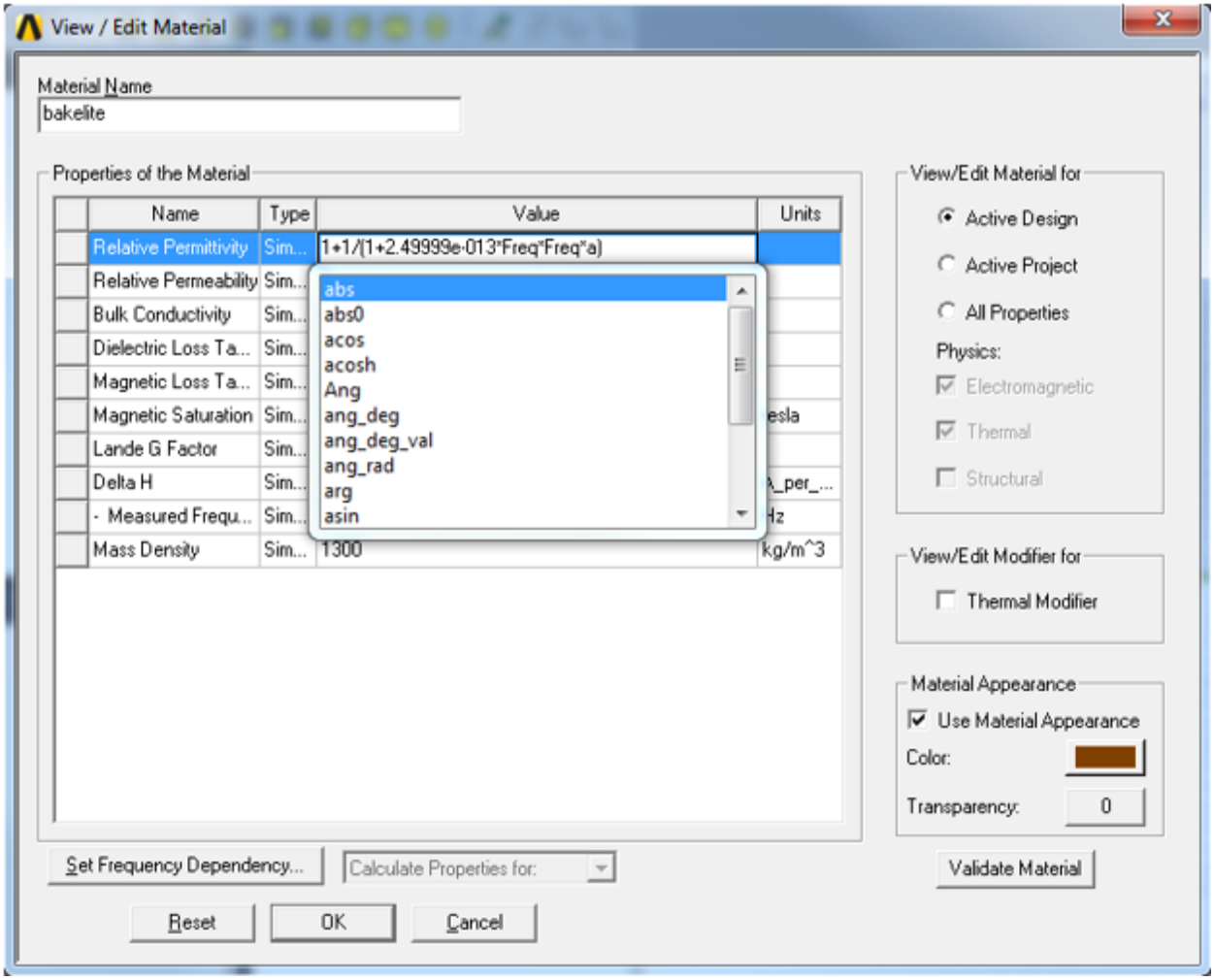
Pressing **[Tab]** or **[Enter]** accepts the current selection and hides the list window. Otherwise, hitting the escape key **[Esc]** hides the auto-complete list window.

For example, the properties window (shown in the picture above for a box drawn in HFSS) has auto-complete configured for both the Material property choices and the Group property choices. When you click in the Material value cell, the drop-down list is displayed, but as you begin to type text, this list is replaced by the auto-complete list if there are matches. For materials, not only are the choices from the drop-down list available for matching, but the full set of material definitions in all loaded libraries is searched for possible matches.

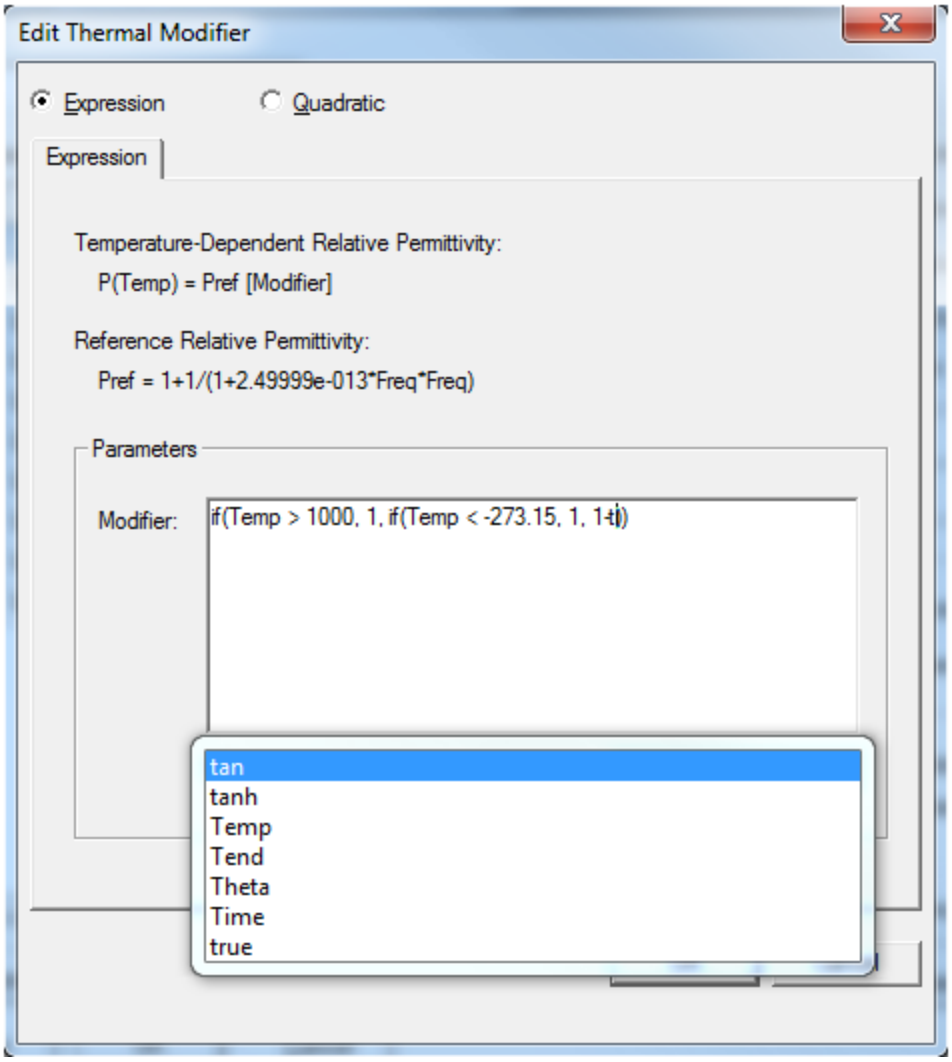
When selecting design properties, auto-complete works with expressions for the values, providing matches for the names of constants, intrinsic variables, functions, project variables (prefixed with “$”), and design properties. When project variables are displayed (that is, you select the project in the Project tree), auto-complete also works for value expressions, but without matching design properties. Properties of other items may also have auto-complete configured to work with value expressions, for example: properties of a CreateBox command or of a circuit component. Below is an example of s design property with auto-complete matching:



Auto-complete also works with value expressions for material properties:



Note the scroll bar, indicating that there are more matches than those currently displayed. Auto complete also works with thermal modifier expressions for Materials.



Related Topics

Showing and Hiding the Properties Window

Setting the Properties Window to Open Automatically

Modifying Object Attributes using the Properties Window

Modifying Object Command Properties Using the Properties Window

Modifying Object Command Properties Using the Properties Window

The **Command** tab in the **Properties** window displays information about an action selected in the history tree that was performed either to create an object, such as the **Draw>Box** command, or to modify an object, such as the **Edit>Duplicate>Mirror** command.

Not all command properties can be modified. The command properties you can typically modify include the numeric values, such as position values (base position, normal position, start position,

etc.), size values (height, radius, etc.), and various other coordinate values. You can also modify many of the unit settings for a command property.

1. In the history tree, select the command for which you want to edit its properties.

Hint	Press and hold **Ctrl** to select multiple commands. If you select multiple commands, only the common, or shared, properties are displayed under the **Command** tab.

2. Under the **Command** tab in the **Properties** window, edit the command's properties.

   Depending on the property type, you can edit it by doing one of the following:

   - Select the check box to apply the property; clear the check box to disable the property.
   - Click in the field and edit the numeric values or text, and then press **Enter**.
   - Click the button and then edit the current settings in the window or dialog box that appears.
   - Click the attribute, and then select a new setting from the menu that appears.

Related Topics

Modifying Objects

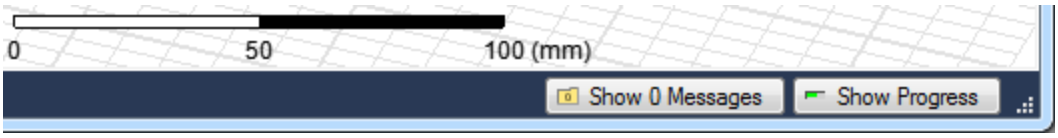
Modifying Object Attributes using the Properties Window

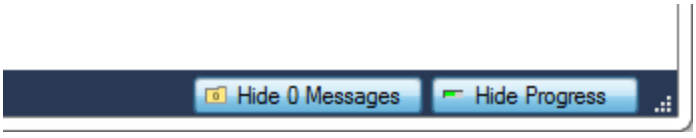
Working with the Progress Window

The **Progress** window monitors a simulation while it is running.

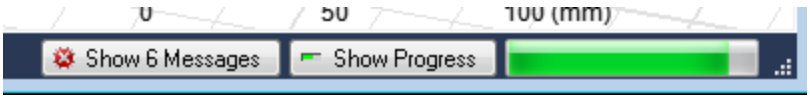
To display or hide the **Progress** window, do one of the following:

- Click the **Show Progress** or **Hide Progress** buttons on the status bar:





When more than one progress bar is active, the top progress bar is represented on the status bar with a progress indicator.



- Click **View> Progress Window**.

  A check box appears next to this command if the **Progress** window is visible.

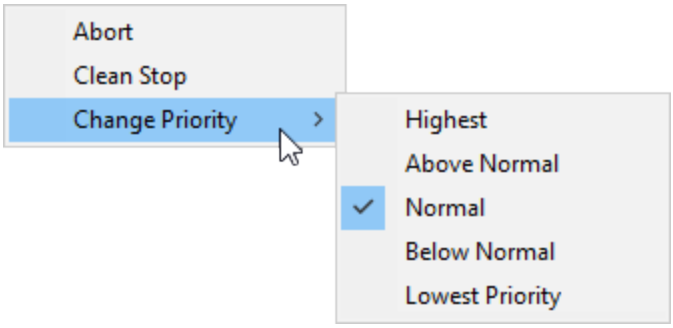
- Right-click the history tree, and then click **Progress** on the shortcut menu.
  A check box appears next to this command if the **Progress** window is visible.

Related Topics

Stopping or Aborting Simulation Progress
Viewing Distributed Analysis Subtasks

Stopping or Aborting Simulation Progress

- To abort progress, right-click in the Progress window, and select **Abort**.
- To stop the simulation cleanly between time steps, right-click in the **Progress** window, and select **Clean Stop**.
- To change priority, right-click. From the **Change Priority** submenu, select from **Highest**, **Above Normal**, **Normal**, **Below Normal**, and **Lowest**.



Stopping and Restarting Time Decomposition Method Analyses

- For Time Decomposition Method (TDM) analyses, invoking **Clean Stop** stops the analysis after completion of the current time subdivision, instead of after completion of the current time step. As a result, continuation of the solution process will start from the next time subdivision after clicking **Analysis**. In such a case, the stop time cannot be modified when TDM is used.
- When **Clean Stop** is invoked for projects in which a TDM transient source design is linked to a transient target design, after completion of the source design the solution process can continue from the target design. In such a case, the target design must be non-TDM.

Viewing Distributed Analysis Subtasks

While a distributed analysis is running, you can access parent and child progress bars. By default, only the main progress bar is displayed, while the child progress bars (or subtasks) remain hidden. You can toggle between showing and hiding the child progress bars.

To show the child progress bars:

- Right-click the progress window, and select **Show Subtask Progress Bars**.

To hide the child progress bars:

- Right-click the progress window, and select **Hide Subtask Progress Bars**.

Working with the Message Manager

The **Message Manager** displays messages associated with a project's development, such as error messages about the design's setup or informational messages about the progress of an analysis.

Messages in the **Message Manager** window are organized first by project, then by circuit. Because a design can contain multiple circuits and subcircuits, sometimes with multiple analyses for each, this organization helps you to quickly determine where errors have occurred. The following icons appear next to a message to indicate information, warnings, errors, or actions:

	Indicates an informative message.
	Indicates a warning message that may require your attention.
	Indicates an error message that may require your attention.
	Indicates the existence of an action that is associated with the message. Click on the message to invoke the action (the cursor will change to a hand icon when it is placed over the action message).

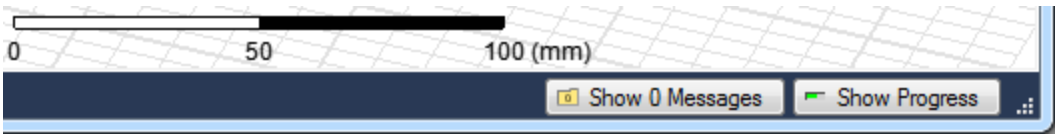
The Message Manager window is a dockable window similar to the Project Manager window. Like the Project Manager window, it can be moved and sized as needed. It can also be attached (docked) to any edge of the desktop. Message text wraps within the window. A vertical scrollbar displays as needed allowing you to move through the list message-by-message.

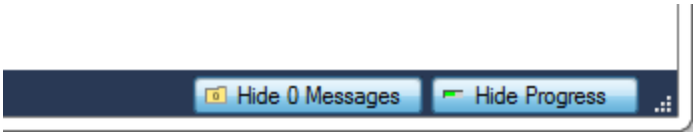
- You can drag the window by its title bar to undock, move, and dock it.
- You can resize the window by dragging its edges.
- You can maximize or minimize the window by clicking the triangle in the title bar.
- You can close the window by clicking the “X” in the title bar.
- You can right-click on a message and select **Details** to show message details.

Note	If a message is too large to fit in the window, clicking the scrollbar to page-up or page-down will have no effect. To view such messages in their entirety, you can resize the window or right-click on the message and select **Details** to show message details.

To display or hide the Message Manager:

- Click **View>Message Manager**.
  A check box appears next to this command if the Message Manager is visible.
- Click the **Show Messages** or **Hide Messages** buttons on the Status bar. The number of messages currently in the Message Manager is also shown in the button.

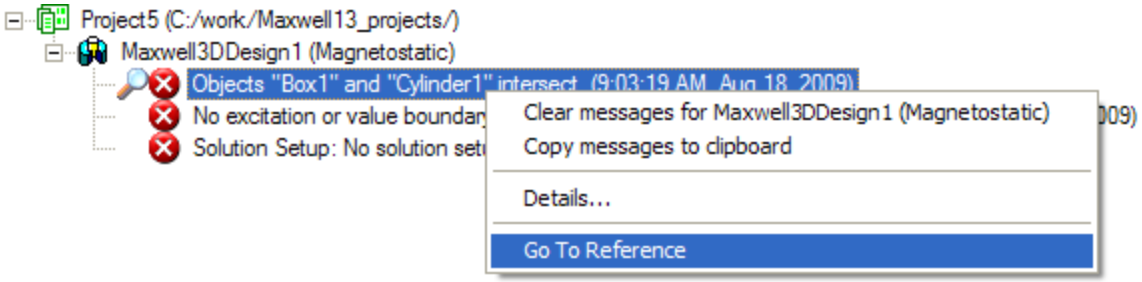




If you right click in the **Message Manager** window, you see a popup menu that lets you:

- Clear the messages for the current model.
- Copy the messages to the clipboard. This can be helpful for sending the messages to application engineers.
- Details. This brings up a information dialog with the project and design for specific message.

After you run a validation check, you can right-click on an intersection error message in the Message window, and select **Go to reference** from the shortcut menu. This selects the intersecting objects.



Related Topics

Showing new messages

Showing errors and warnings

Setting the Message Manager to Open Automatically

Setting the Message Manager to Open Automatically

You can set the Message Manager to open automatically to show new messages and errors and warnings.

Related Topics

Showing new messages

Showing errors and warnings

Showing new messages

You can set the **Message Manager** to automatically be brought up when a new message appears.

1. Click **Tools>Options>General Options**.

   The **General Options** dialog box appears.

2. Under **General>Interface Options**, select **Show Message Window on new messages**.
3. Click **OK**.

Showing errors and warnings

You can set the **Message Manager** to automatically expand when an item is added to a project.

1. Click **Tools>Options>General Options**.

   The **General Options** dialog box appears.

2. Under **General>Interface Options**, select **Ensure that new messages are visible in the Message Window Tree**.
3. Click **OK**.

Clearing Messages for the Project

You can clear all the messages for a particular project.

To clear messages:

1. Right-click the *project#* in the **Message Manager**.

   A pop-up appears.

2. Click **Clear messages for Project#**.

Clearing Messages for the Model

You can clear all the messages for a particular model.

To clear messages:

1. Right-click the *MaxwellModel#* in the **Message Manager**.
   A pop-up appears.
2. Click **Clear messages for MaxwellModel#**.

Copying Messages

You can copy all the messages for a particular project.

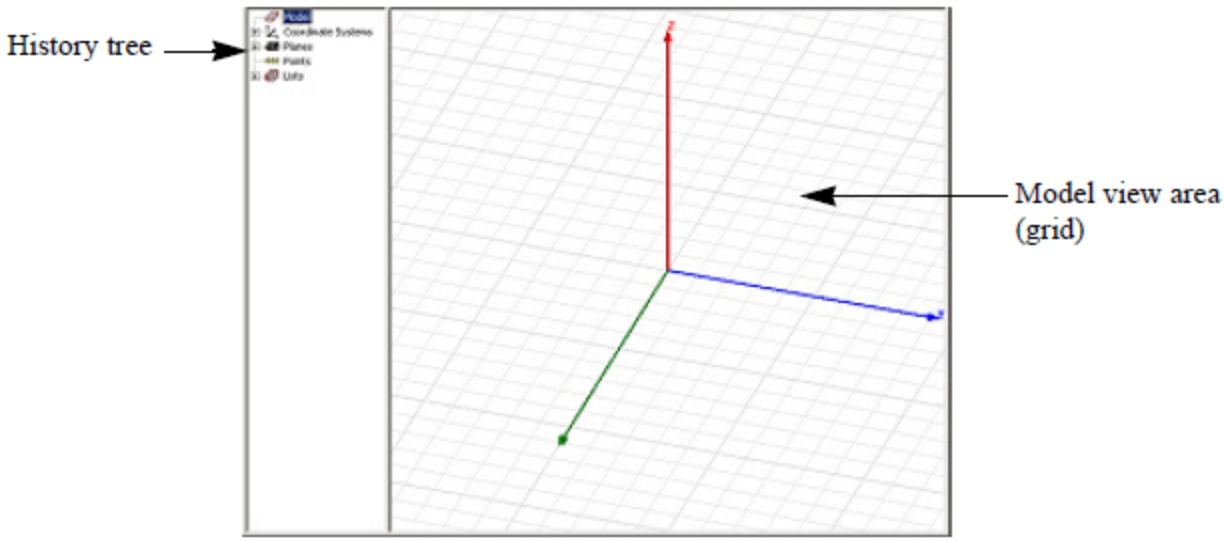
To copy messages:

1. Right-click in the **Message Manager**.
   A pop-up appears.
2. Click **Copy messages to clipboard**.

Working with the Modeler Window

The **Modeler** window is the area where you create the model geometry. It appears to the right of the **Project Manager** window *after* you insert a Maxwell design into a project.

The **Modeler** window consists of the model view area, or grid, and the history tree, as shown below:



To open a new Modeler window, do one of the following:

- Insert a Maxwell design into a project.
- Double-click a Maxwell design in the project tree.

The model you draw is saved with the current project when you click **File>Save**.

Objects are drawn in the **Modeler** window. You can create 3D objects by using Maxwell's **Draw** menu commands, or you can draw 1D and 2D objects and then manipulate them to create 3D objects. For more information, see *Drawing a Model*.

You can modify the view of objects in the **Modeler** window without changing their actual dimensions or positions. For more information, see *Modifying the Model View*.

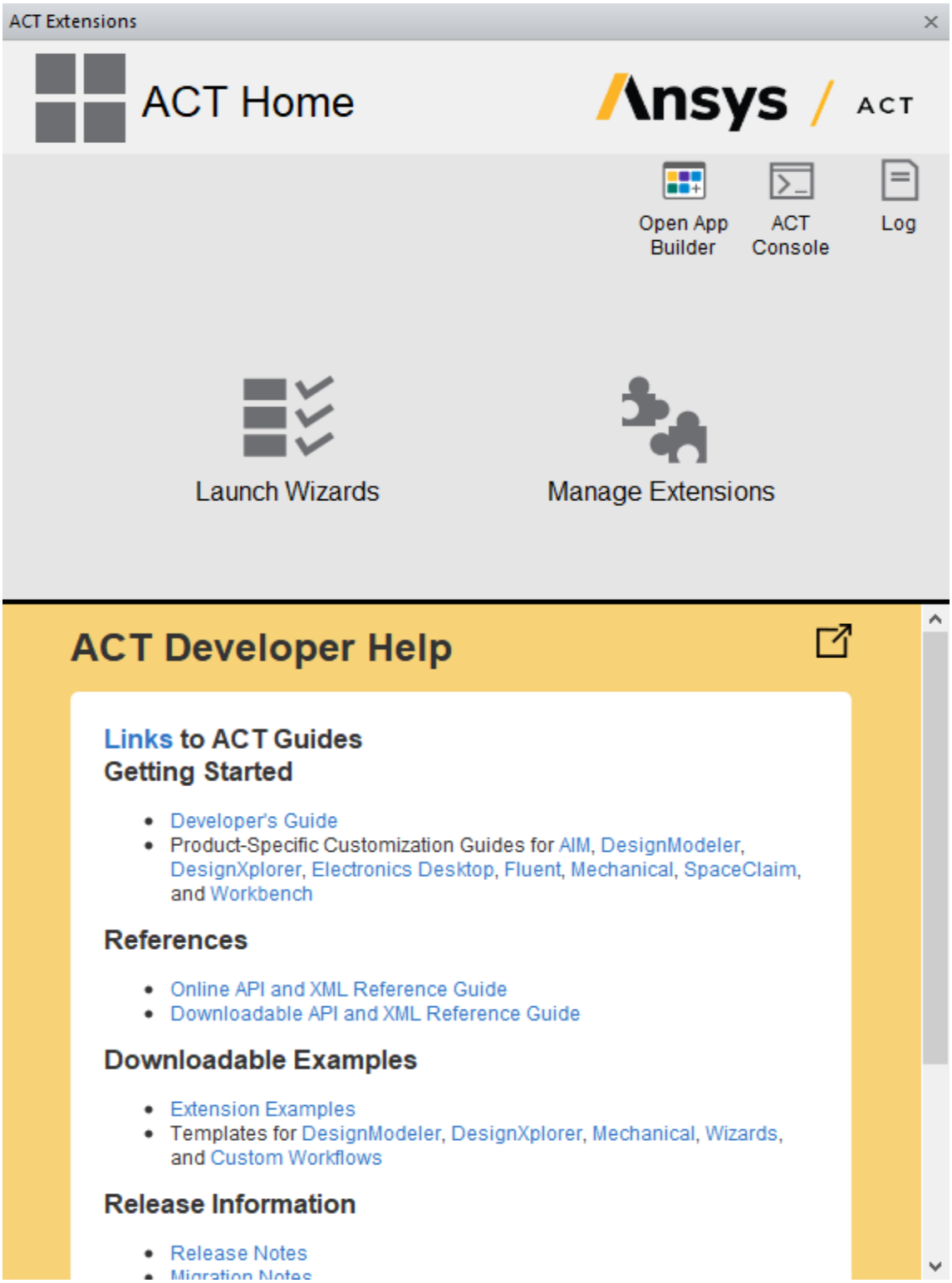
Related Topics

Modifying the Model View

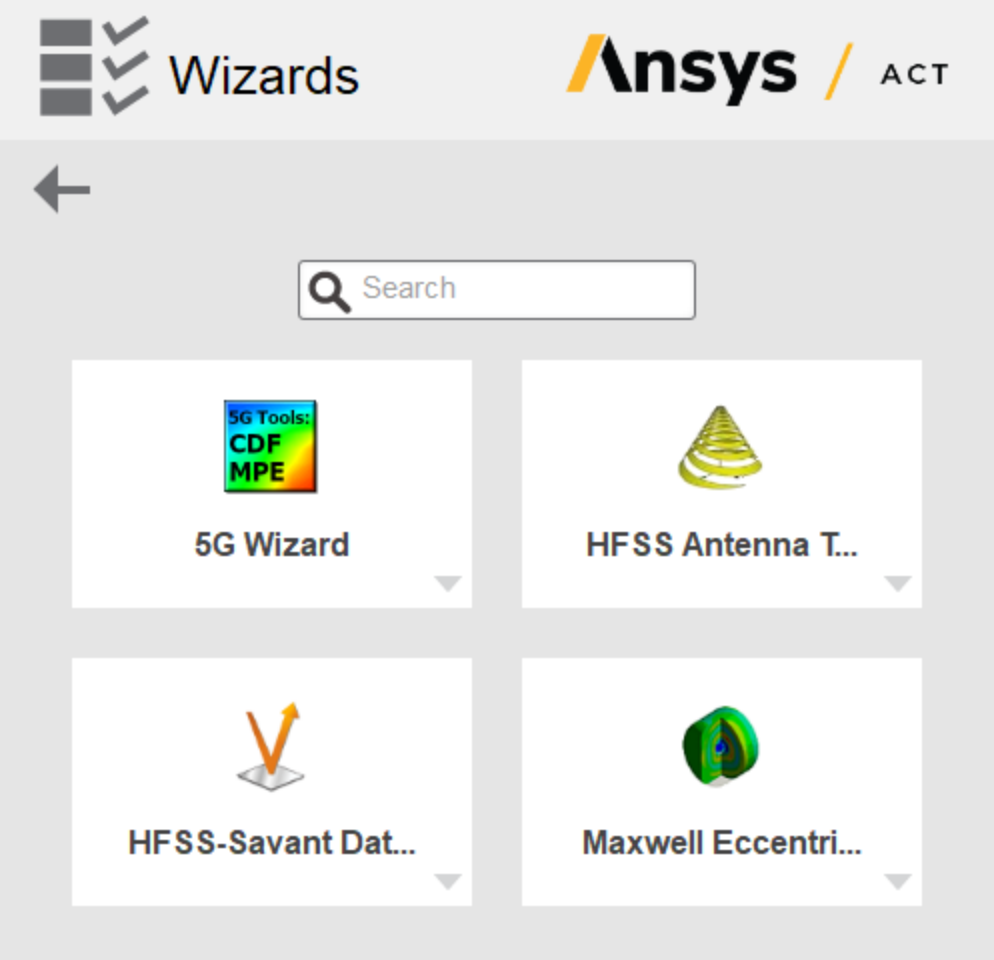
Keyboard shortcuts for the 3D Modeler Window.

Working with the ACT Extensions Window

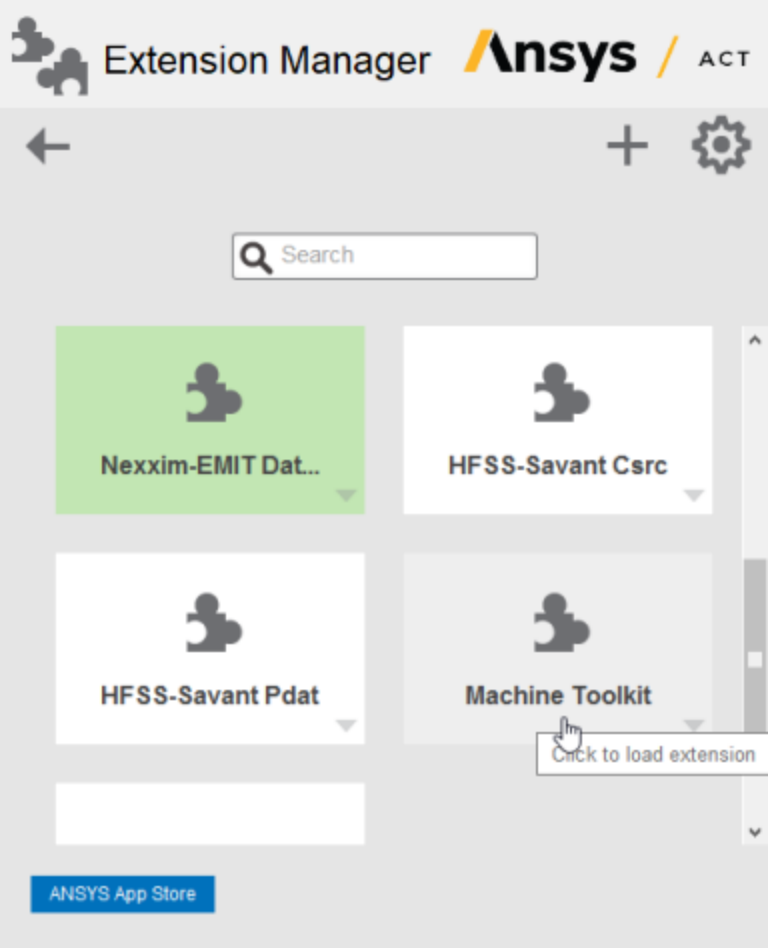
The **View** > **ACT Extensions** command opens the ACT Extensions window for extensions – known as **Wizards** – that have been implemented via Ansys ACT. Depending on how you size and arrange the desktop, the appearance of the window may vary. You may also choose to detach and size it as a floating window to suit your needs.



1. Click on the **Launch Wizards** icon to view active wizards in the **Wizards** view.

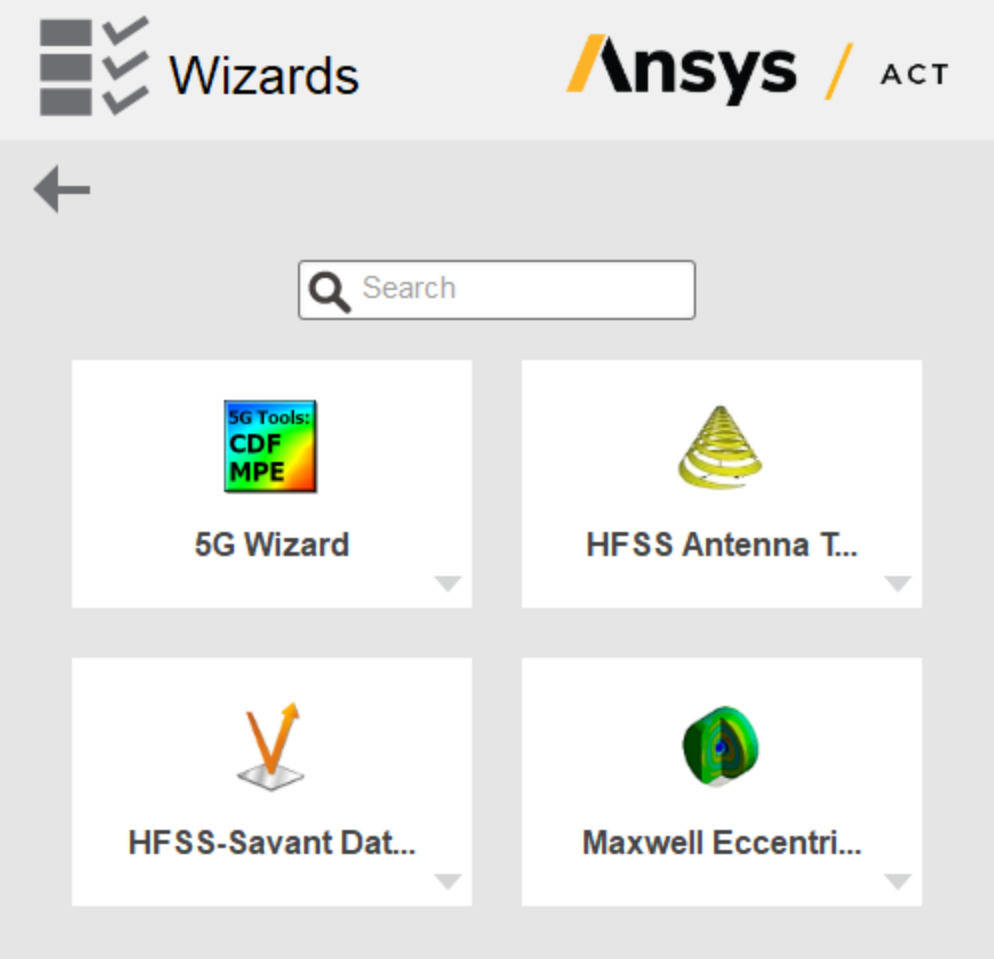


2. To add and use an existing Wizard, you must activate it by clicking on the **Manage Extensions** icon. This changes the view to show the installed extensions. If there are many available extensions, you can either scroll or use the **Search** field to find the ones you want. **Search** is not case sensitive. Click the desired extension to activate it. The extension is then loaded and highlighted to show the change in status. You can also unload (deactivate) an extension by clicking on it. In the following image, the Nexxim extension has been activated, but the HFSS and Machine Toolkit extensions are not activated for use.

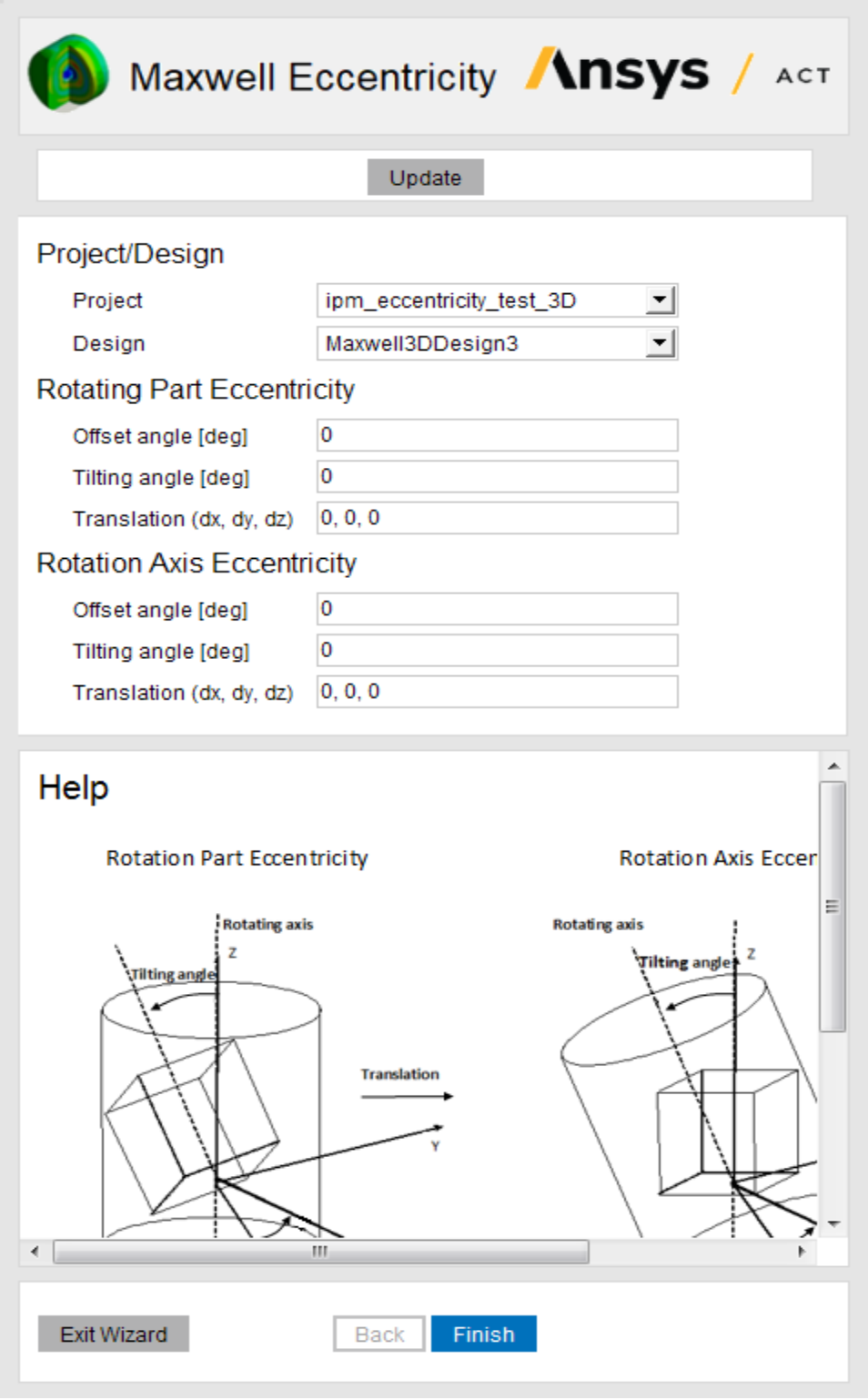


3. Click the back arrow icon to return to the ACT start page,

then click the **Launch Wizards** icon to open the **Wizards** view.



4. Click on the icon for the wizard you want to use to open it in the window.



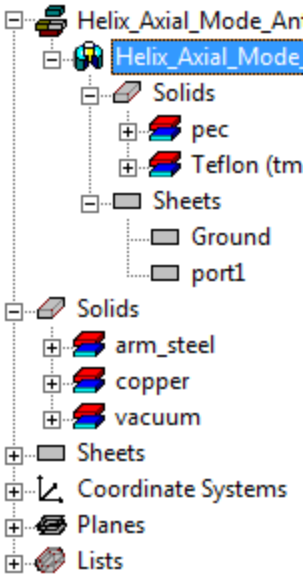
5. You can click the **Exit Wizard** button to exit the wizard and return to the Wizards view.

Related Topics

"Modeling Maxwell Motion Eccentricity " on page 13-9

Working with the History Tree

The history tree in the **Modeler** window lists the active model's structure and grid details, including the model's objects, coordinate systems, default planes, and any object or face lists.



You can perform the following tasks with the history tree.

- Expand or Collapse Groupings in the History Tree
- History Tree Layout Commands
- Group Commands for Modeler Objects
- Selecting Objects in the History Tree
- Use Shortcut Menus in the History Tree
- View Properties for History Tree Objects
- History of Commands on History Tree Objects
- Control the View of Objects in the History Tree
- Selecting All Objects in a History Tree Folder
- Upgrade Version

The history tree contains the following model details:

Invalid	Lists all invalid objects
3D Component Names	Lists any 3D components added to the design, listing their Solids, Coordinate Systems, and Planes, separately from objects drawn in the modeler, or imported to the modeler.

Solids	Displays all the model's solid objects and a history of the commands carried out on each object.
Sheets	Displays all the sheets in the model 3D design area. By default, all sheet objects are grouped by boundary assignment.
Lines	Displays all line objects included in the active model. See *Drawing a Line* for information on how to draw a line object.
Points	Displays all point objects included in the active model. See *Drawing a Point* for information on how to draw a point object.
Groups	Displays folders for Groups you have created or imported. See Group Commands for Modeler Objects.
Coordinate Systems	Displays all the coordinate systems for the active model. See *Setting Coordinate Systems* for more information on this model detail.
Planes	Displays the planes for all the coordinate systems. When you create a coordinate system, default planes are created on its xy, yz, and xz planes.
Lists	Displays the object or face lists for the active model. By default, a list called "AllObjects" appears. Creating an object list is a convenient way to identify a group of objects for a field plot or calculation. Creating a face list is a convenient way to identify a specific set of surfaces for a field plot or calculation. See *Creating an Object List*.

Note	While objects created in Maxwell can always be classed in the history tree as either a solid, sheet, or wire, some imported objects may have mixture of these. Such objects are placed in an Unclassified folder in the history tree.

Expand or Collapse Groupings in the History Tree

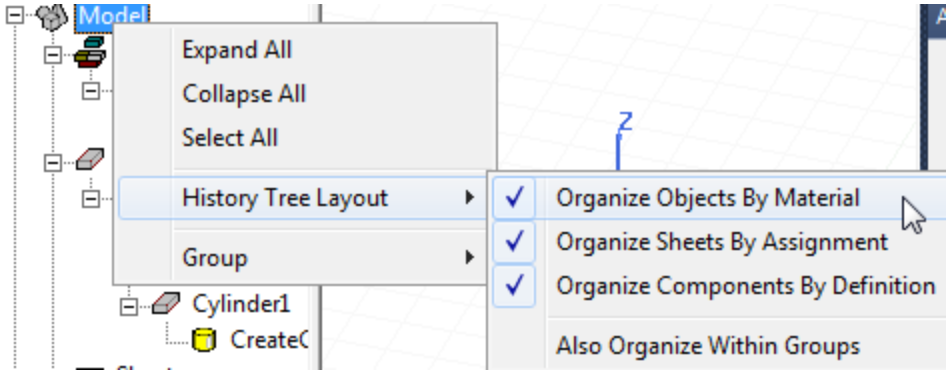
You can expand or collapse object groupings in the history tree by left clicking the + or - respectively. In addition, right-clicking on any group icon opens a pull-down to **Expand All** groupings or **Collapse All** groupings.

Related Topics

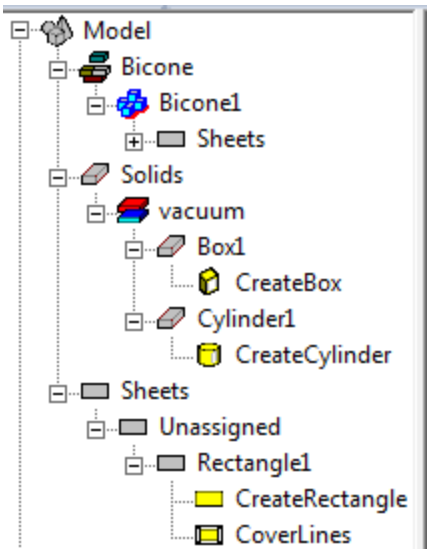
Selecting Objects in the History Tree

History Tree Layout Commands

A set of History Tree Layout commands lets you organize objects by material, assignment, or by component definition. By default there is no further organization within groups. You can select **Also Organize Within Groups** to extend organize settings within groups.



If the design does not contain material assignments, sheets, components, or groups, the menu items are disabled. You can access the History Tree Layout commands by clicking **Modeler>History Tree Layout** or by right-clicking on the Model icon in the history tree (as shown above). You click on [-] to close a hierarchy and [+] to open one.



Related Topics

Select Objects in the History Tree

Group Commands for Modeler Objects

History of Commands on Objects

The history tree also lists the history of all commands carried out a model's objects. This history is displayed in the order in which it occurred. Note in the above image the expanded air object and

its history of commands.

Selecting a command in the history highlights the object in the Modeler window and shows that object's properties (if available) in the docked Properties window. You can look at the fields in the Properties window to see any editable fields for that command, such as coordinate system, line type, coordinates, or units.

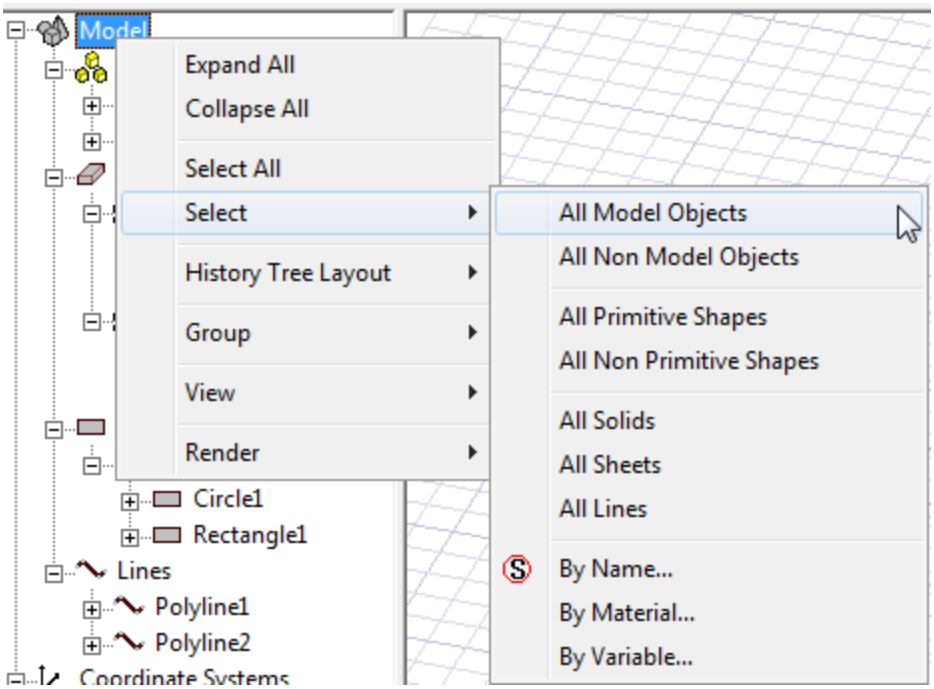
For some commands, such as **Edit>Arrange>Move**, or boolean operations, selecting them in the history tree enables the X (Delete) icon on the ribbons and the **Edit>Delete** menu. In these cases, you can delete those commands from the history tree as a way of undoing those operations. As an alternative to deletion, you can check **Suppress** command in the Properties window for that command. This undoes the effect of a command on an object without removing it from the History tree.

Selecting Objects in the History Tree

Selecting objects in the History tree also selects them in the View window. This can be useful for complex objects, when it may be easier to find the objects of interest by name or material, if the object of interest is inside or behind others.

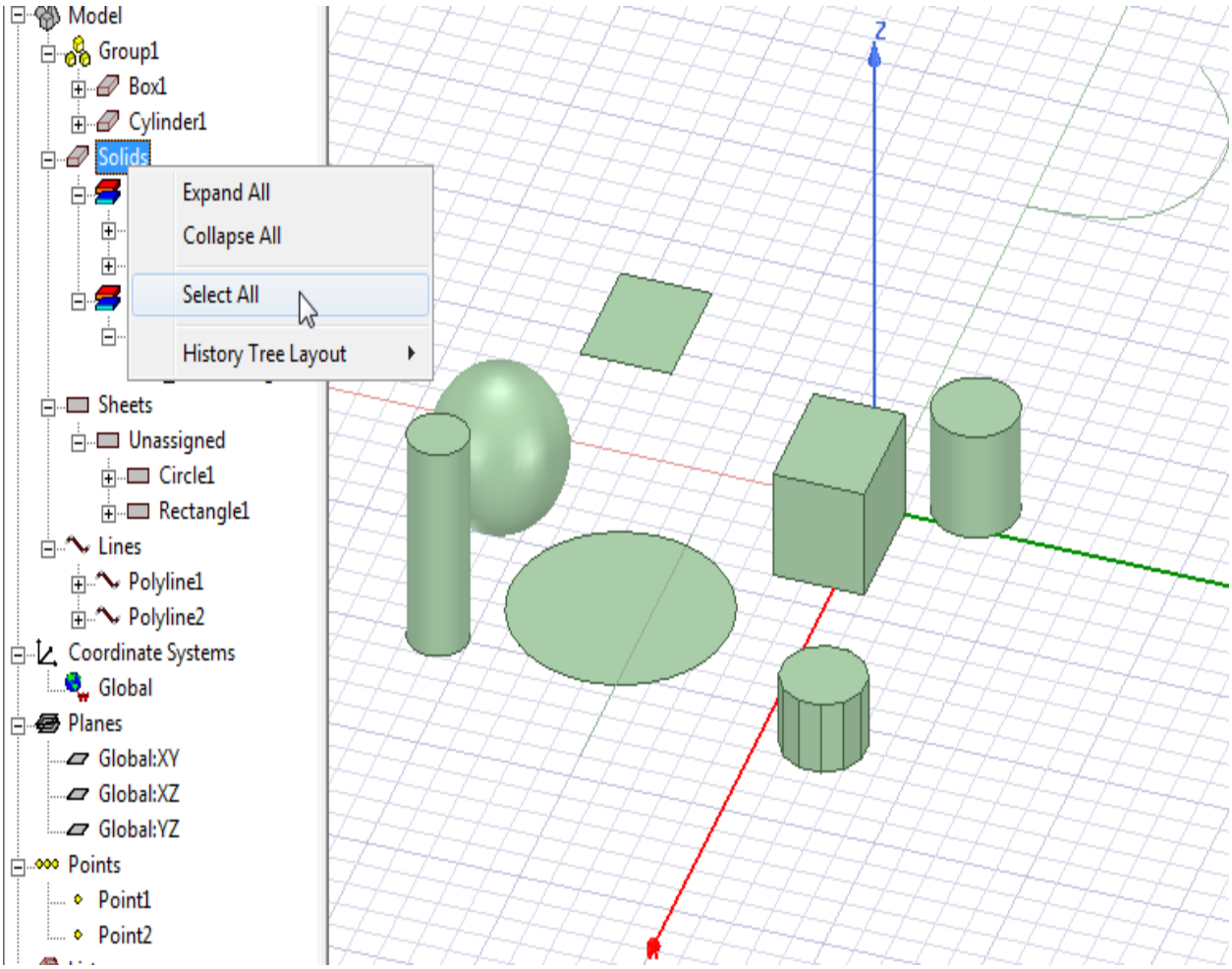
You can use CTLR-click to make multiple selections. You can select a range of objects by a click on the first, and then SHIFT-Click to select all in the range. You can also click and drag the mouse to make rubber band selections. Only visible objects are selected. That is, if the hierarchy is closed under the selection, any operand parts are ignored and do not interfere with cut and paste operations.

If you select Model in the History tree and right-click the shortcut menu offers these choices:

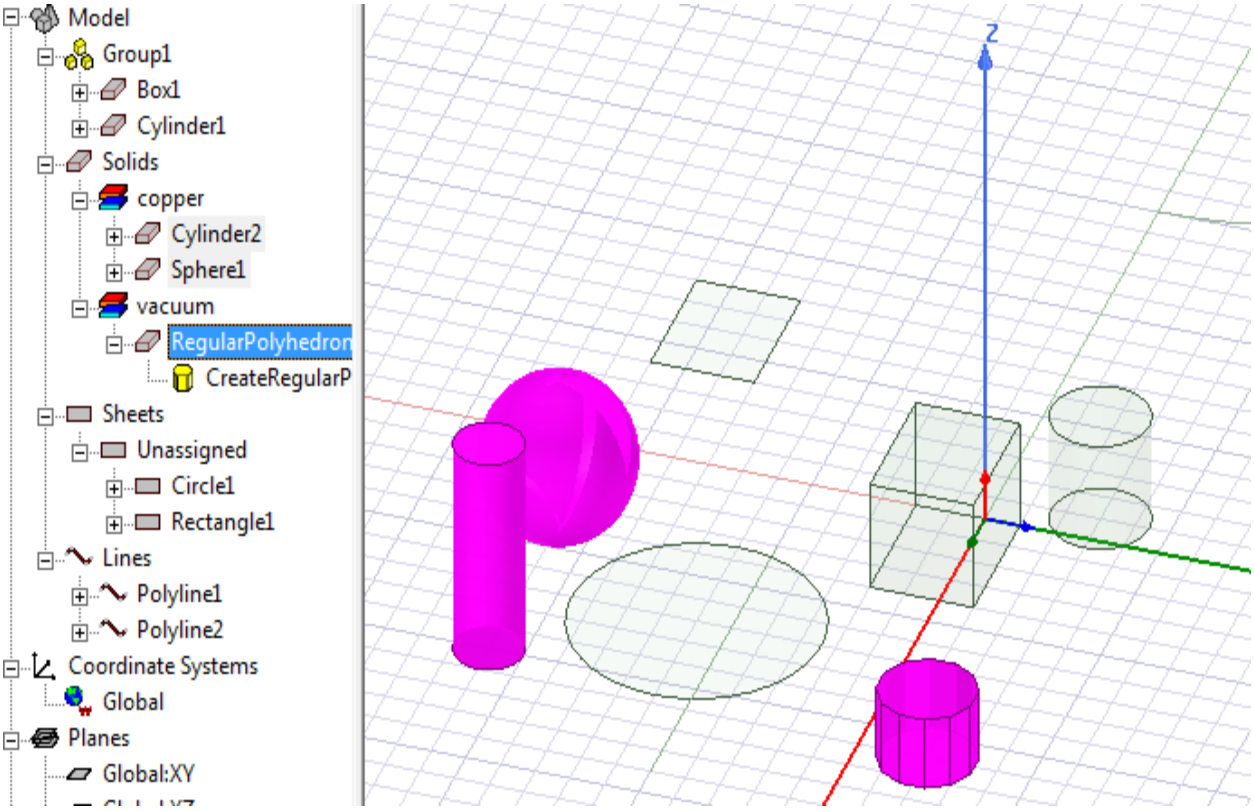


You can also right click on any History tree object or grouping and right click to display a shortcut menu with a **Select All** command. This lets you select at one time all solids, sheets, lines, non-model objects, or all unclassified objects, as well as all non-grouped objects assigned to specific materials.

For example, right-click on Solids and then **Select All**:



After clicking **Select All**, the History tree and the **Model** window highlight the selected Solid Objects. Notice that Box1 and Cylinder1 are not selected because they are in Group1.



If you select a geometric object, such as a Box or Cylinder, the shortcut menu **Select** command offers choices to select **All Faces**, **All Edges**, or **Sheet Edges** for that object.

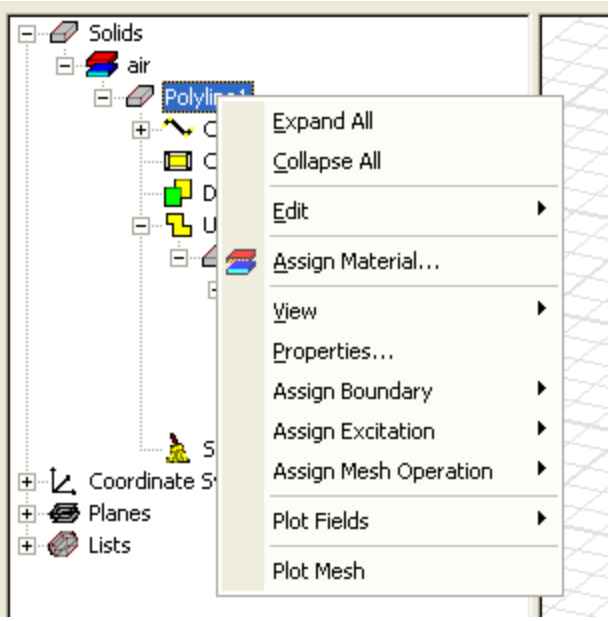
Related Topics

History Tree Layout Commands

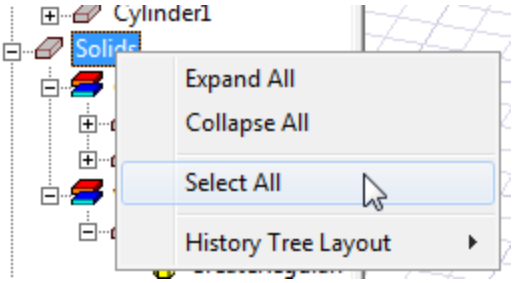
Group Commands for Modeler Objects

Shortcut Menus in the History Tree

If you select particular objects in the history tree, right-click displays a shortcut menu lists the commands that you can apply to the selected command, object, or objects. The shortcut menus for model objects are most extensive.

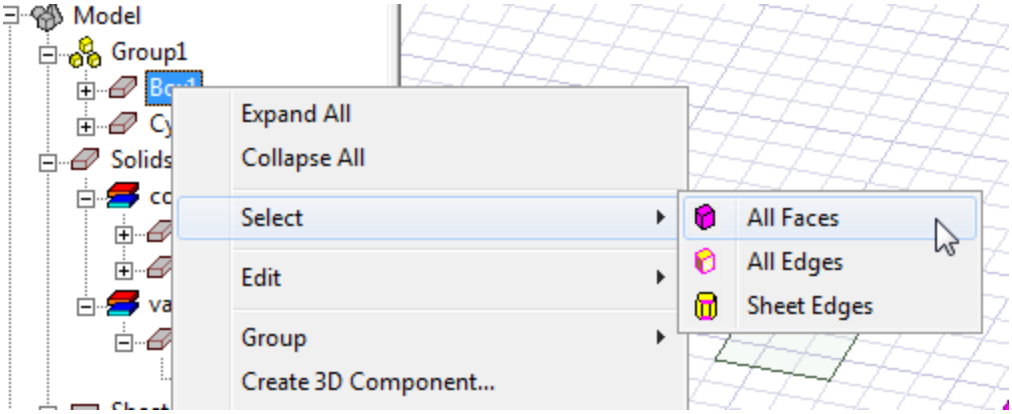


If you right click on a model or category such as Solids, Sheets, Lines, or a material in the History tree, the short cut menu includes a **Select All** command.

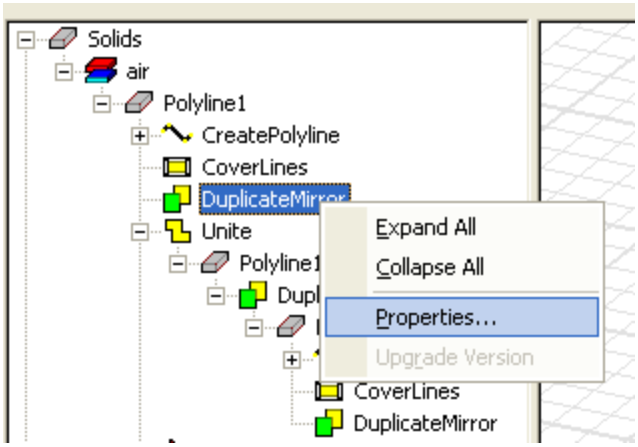


The behavior of the shortcut **Select All** command depends on the History tree model, object or property icon from which you invoke the shortcut menu. For explanation and examples, see Select Objects in the History tree.

If you select an object in the History tree, the menu offers selections for that object.



In other cases, you can only view properties, or expand or collapse hierarchy.



Related Topics

View Properties for History Tree Objects
History of Commands on History Tree Objects

View Item Properties in the History Tree

To view the properties of an item in the history tree:

- Click the item's name in the history tree.

  The item's properties appears in the docked **Properties** window.
- Double-click on an item in the history tree to display a **Properties** window.
- Click the item's name in the history tree, and double right click to display a shortcut menu. Then select **Properties** to display the **Properties** window.

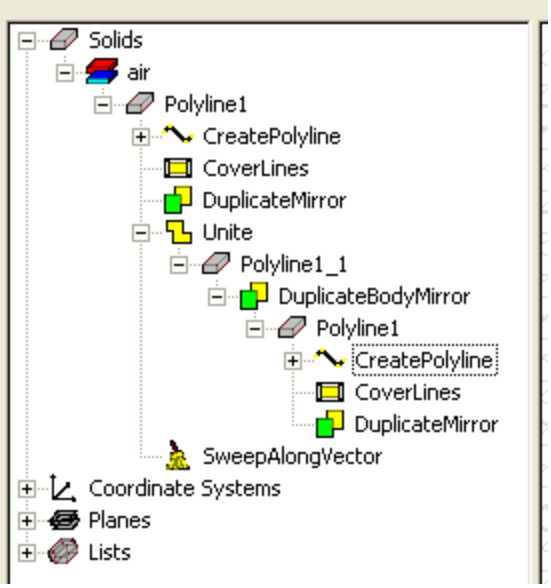
Related Topics

Purge History command

Generate History command

View and Edit Commands on History Tree Objects

The history tree also lists the history of all commands carried out a model's objects, for example, "CreateBox" or "Subtract." This history is displayed in the order in which it occurred. Here is a history tree from the waveguide combiner example.



Selecting a command in the history highlights the object in the Modeler window and shows that object's properties (if available) in the docked Properties dialog box. You can look at the fields in the Properties dialog to see any editable fields for that command, such as coordinate system, line type, coordinates, or units.

For some commands, such as **Edit>Arrange>Move**, or boolean operations, selecting them in the history tree enables the X (Delete) icon on the ribbons and the **Edit>Delete** menu. In these cases, you can delete those commands from the history tree as a way of undoing those operations. As an alternative to deletion, you can check **Suppress** command in the Properties window for that command. This undoes the effect of a command on an object without removing it from the History tree.

Related Topics

Select Objects in the History Tree

Purge History command

Generate History command

Modifying Objects

Controlling the View of Objects in the History Tree

To control the view and visibility of an object such as a box or PML, right click on an object in the history tree display the short-cut menu and select **View**. The short cut menu contains the following commands:

- Fit in Active View
- Hide in Active View
- Show in Active View
- Fit in All Views
- Hide in All Views
- Show in All Views

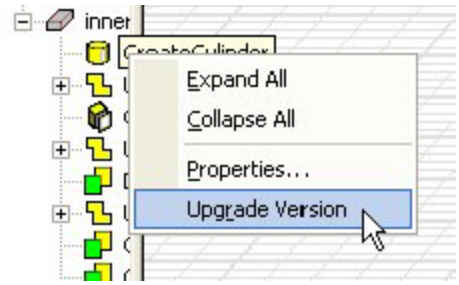
Related Topics

Purge History

Generate History

Upgrade Version in History Tree Shortcut Menu

By right-clicking on an operation icon in the history tree in the Modeler window, you can see the **Upgrade Version** command.



By default, the Modeler opens legacy projects using the modeling kernel version under which the project was saved, so that users don't see side effects from slight topology changes between versions. The **Upgrade Version** operation forces Electronics Desktop to use the latest modeling kernel. Typically, this would only by used when a needed fix is available with the newest version.

Related Topics

Working with the History Tree

Keyboard Shortcuts for the 3D Modeler Window

The following keyboard shortcuts apply to the 3D Modeler Window

- **B**: Select face/object behind current selection
- **F**: Select faces mode
- **O**: Select objects mode
- **E**: Select edges mode
- **V**: Select vertices mode
- **M**: Mult select mode
- **Ctrl + A**: Select all visible objects
- **Ctrl + Shift + A**: Deselect all objects
- **Ctrl + D**: Fit view
- **Ctrl + E**: Zoom in, screen center
- **Ctrl + F**: Zoom out, screen center
- **Shift** + LMB: Zoom in / out
- **Alt** + LMB: Rotate model
- **Alt + Shift** + LMB: Zoom in / out
- **Alt** + Double-click LMB: Sets model projection to standard isometric projections (cursor must be in corner of model screen N/NE/E/SE/S etc)
- **F3**: Switch point entry mode to mouse (draw objects by mouse). This switch takes effect only after the drawing of an object has started.
- **F4**: Switch to dialog entry mode (Draw objects by entry in the command and attribute box). This switch takes effect only after the drawing of an object has started.
- **F6**: Render model wire frame
- **F7**: Render model smooth shaded

Note	LMB means Left Mouse Button.

To customize the shortcut assignments, use Tools>Keyboard Shortcuts.

Related Topics

General Purpose Keyboard Shortcuts for Maxwell

Custom Keyboard Shortcuts

Running Ansys Electronics Desktop From a Command Line

Ansys Electronics Desktop includes line arguments that can be included when launching from a command line or terminal prompt. All command-line arguments are case-insensitive. The commands associated batch options can also be used with a Job Management Interface for submitting jobs to Ansys or RSM and other supported schedulers. See

RSM Integration with Job Management UI

Integration with Microsoft Windows® HPC Scheduler

Integration with Platform's Load Sharing Facility (LSF)

Integration with Grid Engine (GE)

Command-line syntax

```
ansysedt <options> <run command(s)> <project file/script file>
```

It is good practice to put quotation marks around the path to the solver executable, and around the full path to the project. This ensures that spaces in the path or project will not be an issue. The same is true of the design name, if there are spaces.

Run Commands

The following command line run commands are available for Ansys Electronics Desktop. Of the commands (BatchSave, BatchSolve, BatchExtract, RunScript, RunScriptandExit), one or none must be specified as arguments after the solver executable. When none is specified, you may specify a project or archive to open when Electronics Desktop launches, and can only use the **-Iconic** and **-Help** options. The commands are further described below:

- **-BatchSave** [options] *file or folder specifier***>**

  Saves a named project or folder containing one or more project files or folders to the current version. This can be helpful for the change from ACIS to Parasolid. The Parasolid migration of aedt projects does not overwrite the existing project. Instead a copy is created through save-as with _converted suffix and then that copy is migrated to Parasolid. You can run this command with the **-Iconic**, **-Logfile**, and **-ng** options.

- **-BatchSolve <***project filename***>**

  By default, this run command solves all adaptive setups, Optimetrics setups, and sweeps found in the project file. You can run this command with the **-Iconic**, **-Logfile**, **-ng**, and **-WaitForLicense** options. If parallel solve is possible, you can use the **-Distribute** option.

  If you wish to specify which setups **-BatchSolve** completes, you can use additional parameters:

  - **[*designName*]** – batch solve all setups for the specified design in the project file.
  - **[*designName*]:Nominal** – batch solve all nominal setups for the specified design in the project file.
  - **[*designName*]:Optimetrics** – batch solve all Optimetrics setups for the specified design in the project file.
  - **[*designName*]:[*Nominal/Optimetrics*]:[*SetupName*]** – batch solve the specified Nominal or Optimetrics setup in the specified design.

  If you wish to specify whether **-BatchSolve** setups are completed locally or remotely, you can use the following options:

  - **-Local** – performs the **-batchsolve** on the local machine.
  - **-Remote -machineList** – performs the **-batchsolve** on a remote machine. The <*machineList*> should provide a single hostname.

- **-Distributed -machineList** – performs a distributed **-batchsolve** using a specified machine list.

  The **-machineList** parameter for a **-Distributed** setup can be formatted three ways:

  - **-MachineList list="<***machine1***>, <***machine2***>, ..."** – machine names (either by IP address or hostname) are separated by commas. If the list contains any whitespace, it must be enclosed in quotes. The number of distributed COM engines run on each host is equal to the number of times the hostname appears in the list. That is, if host1 appears in the list once, and host2 appears twice, then one COM engine will run on host1 and two COM engines will run on host2.

    list= accepts the following additional modifiers:

    <*MachineName***>:<***NumTasks***>:<***NumCores***>:<***RAM Limit in %***>:<***Number of GPU***>**

    Example:

    ```
    list="Orion:4:8:90%:1, Aries:3:12, Pluto:6:12"
    ```

    > **Note:**
    >
    > Duplicate machines are not allowed when specifying these additional modifiers. The number of cores must be greater than the number of tasks.

  - **-MachineList file="<***machineListFilepath***>"** – machine names (either by IP address or hostname) are listed in a file (one per line), and you specify the filepath. The number of distributed COM engines run on each host is equal to the number of times the hostname appears in the file. That is, if host1 appears in the file once, and host2 appears twice, then one COM engine will run on host1 and two COM engines will run on host2.

    file= accepts the following additional modifiers, in the file itself:

    <*MachineName***>:<***NumTasks***>:<***NumCores***>:<***RAM Limit in %***>:<***Number of GPU***>**

    Example:

    ```
    "Orion:4:8:90%:1",
    "Aries:3:12",
    "Pluto:6:12",
    ```

> **Note:**
>
> Duplicate machines are not allowed when specifying these additional modifiers. The number of cores must be greater than the number of tasks.

- **-MachineList num="<***numberofDistributedEngines***>"** – This format is used when a scheduler (such as LSF, PBS, SGE or HPC) is used to manage the jobs sent to a cluster of hosts. In a scheduler environment, you can specify the number tasks for distributed processing. In this case, you do not specify the machine names after the flag because the names are provided by the scheduler. For example, in the Windows HPC environment, you can write the number of tasks as follows:

```
-MachineList num=4
```

-Distributed setups can also take the following optional arguments. When these are not present, the behavior defaults to single-level distributed solutions with no change in order of precedence among possible distribution types.

The arguments are:

- **includeTypes=<***default***>|<***distribution type 1, distribution type 2***, ...>** – If included distribution types are specified, only the listed distribution types are enabled. If default is specified, the default set of enabled distribution types is used. To see a valid distribution types for your design, click **Simulation** > **Analysis Config** to open the **Analysis Configuration** window and view the types on the **Job Distribution** tab.
- **excludeTypes=<***default***>|<***distribution type 1, distribution type 2***, ...>** – If excluded distributed types are specified, all distribution types except those listed will be enabled. If default is specified, the default set of enabled distribution types is used. To see a valid distribution types for your design, click **Simulation** > **Analysis Config** to open the **Analysis Configuration** window and view the types on the **Job Distribution** tab.
- **maxLevels=<***1/2***>** – the maximum number of levels of job distribution (the current maximum is 2). See: Selecting Optimal Configurations for Distributed Analysis.
- **numLevel1=** – when two-level distribution is selected (**maxLevels=2**), this specifies the number of level 1 tasks.

You can also specify how a **-BatchSolve** distributes Optimetrics variations:

- **-auto** – Without additional parameters, the batch log file will specify that Optimetrics variations be solved sequentially.
- **-auto NumDistributedVariations=<***num***>** – You can specify an integer value greater than 1. This is the number of variations that will be solved in parallel.

- **-BatchExtract <***BatchExtract script filename***><***project filename***>** – allows the following commands to be executed non-graphically via script and without checking out any GUI licenses:
  - ExportProfile
  - ExportConvergence
  - ExportMeshStats
  - ExportNetworkData
  - ExportNMFData
  - ExportEigenmodes
  - ExportTransientData
  - Update Reports
  - ExportToFile

  A project file *must* be specified when **-BatchExtract** is used. Commands in the script file will only be executed in the specified project.

  > **Note:**
  >
  > - **-ng** must be used with **-BatchExtract**, or it will fail.
  > - Only the scripts listed above are supported for **-BatchExtract**. Including unsupported script commands will terminate script execution.

- **-RunScript <***script filename***>** – runs the specified script. You can use the **-ScriptArgs** option to add one or more arguments to this command, and the **-Iconic** option.
- **-RunScriptAndExit <***script filename***>** – runs the specified script and exits Electronics Desktop. You can use the **-ScriptArgs** option to add one or more arguments to this command. You can also use **-Iconic**, **-Logfile**, and **-WaitForLicense**.

  > **Note:**
  >
  > **-BatchSolve <***DesignName***>** is mutually exclusive with **-RunScriptAndExit <***ScriptName***>**.

- **-monitor** – during non-graphical analysis, you can monitor progress and messages. Progress, warning and info messages are logged to the standard output stream. Error and fatal messages are logged to the standard error stream. Schedulers intercept these streams and provide commands for display of this output. See individual scheduler documentation for specifics.

Job Management from the Command Line

- **-showmonitorjob** – Launch the monitor job dialog.
- **-showsubmitjob** – Launch the submit job dialog.

- **-showselectscheduler** – Launch the select scheduler dialog.

Run Command Examples

A distributed **-BatchSolve** of a specified design's Optimetrics setups, with a specified machine list:

```
C:\Program Files\AnsysEM\v231\Win64\ansysedt -distributed -
machinelist list="255.255.1.1,255.255.1.2" -batchsolve
myDesign:Optimetrics
"%UserProfile%\Documents\Ansoft\myProject.aedt"
```

A **-BatchExtract** operation using paths to a script file and a project file:

```
ansysedt -ng -batchextract exportToFile.py "C:\Program
Files\AnsysEM\AnsysEM23.1\Win64\Examples\ElectronicsDesktop\HFS
S\RF Microwave\OptimTee.aedt"
```

Where exportToFile.py contains:

```
oDesktop.RestoreWindow()
oProject = oDesktop.SetActiveProject("OptimTee")
oDesign = oProject.SetActiveDesign("HFSSDesign1")
oModule = oDesign.GetModule("ReportSetup")
oModule.UpdateReports(["XY Plot 1"])
oModule.ExportToFile("XY Plot 1", "exportToFilePy.csv")
```

A **-BatchSolve** of a specified design's nominal setups, run in a minimized window, with a specified log file:

```
ansysedt -Iconic -LogFile "H:\Logs\mylog.log" -BatchSolve
myDesign:Nominal "H:\Projects\MyProject.aedt"
```

Specifying Project Files

Specifying a project file opens that project when Electronics Desktop launches. If **-BatchSolve** is set, the project will also be solved.

You can specify an archive file instead of a project file. If **-Batchsolve** is set, the project will be automatically restored and solved. Otherwise, you are prompted for a restore location, and the project will be restored and opened.

When a **-Batchsolve** is being performed on an archive file, you may also specify **-archiveoptions**:

- **overwritefiles** – allows non-project/results-extracted files to overwrite existing files.
- **path=<***projectFilepath***>** – extracts the project file and associated files to the specified path. If not specified, the archive will be extracted into the same directory as the archive file.
- **winpath=<***windowsProjectFilepath***>** – specifies the Windows-specific path to the extracted project file. This is used when a batch job is to be run on a Linux system, but monitored on Windows.

Options

The following options can be associated with one or more of the run commands:

- **-autoextract** – exports profile (as text), convergence (as text), and report data (as CSV) for the requested project/design/setup in a batch job. Once the solve is complete, an export directory is created (for example, "Project1.aedtexport" for a project named "Project1.aedt") that contains a sub-directory for each design name. You can also specify `-autoextract "reports, fieldplots"` to also generate *.aedtplt files for each field plot and possible *.avz file (for import and display in Ensight) for all valid field plots. Export files reside within each design-name directory, and include setup name, design variation, job ID, and problem type, as applicable.

  > **Note:**
  >
  > - The -autoextract option is only valid when used with -BatchSolve
  > - The -autoextract option is automatically added for all Ansys Cloud jobs submitted from Electronics Desktop. There is also an additional "reports" and/or "fieldplots" option that immediately follow "-autoextract". This causes all reports to be exported as CSV files at the end of the batch solve, after the profile and convergence have been exported.
  > - For example, you can specify `-autoextract "reports, fieldplots"` to also generate *.aedtplt files for each field plot and possible *.avz file for all valid field plots.

- **-batchoptions** – for batch jobs, specifies any of the options in **Tools** > **Options**. See additional information.
- **-batchoptionhelp** – opens a window showing -batchoptions help. The paths shown in this window can be used with batchoptions and the Update Registry Get and Set commands. See: Setting or Removing Option Values in Configuration Files: UpdateRegistry Command.
- **-distribute** – distributes a batch solve to multiple machines. This option must be combined with the **-BatchSolve** run command. See: Distributed Analysis.
- **-help** – opens a window displaying command line options. This can only be used without a run command.
- **-iconic** – runs Electronics Desktop with the window iconified (minimized).
- **-logfile<***filePath***>** – specifies a log file. If none is specified, <project_name>.log will be written to the <project_name>.batchinfo directory.
- **-monitor** – enables batch job output to standard output and standard error streams.
- **-ng** – runs Electronics Desktop in non-graphical mode. This *must* be used with the **-BatchExtract** command.
- **>-useElectronicsPPE** – selects Electronics Pro/Premium/Enterprise licensing. Allowed values: <none>, true, "= 1", true, "=0" false.

- **-validateonly** – If true, validation will be performed, but none of the setup will be solved. Allowed values: 0 (false), 1 (true).
- **-waitforlicense** – directs Electronics Desktop to wait for unavailable licenses.
- **-scriptargs** – used in conjunction with **-RunScript** or **-RunScriptAndExit**, adds arguments to the specified script. You can pass multiple arguments to **-scriptargs** by surrounding the arguments in double quotes. For example:

  ```
  ansysedt.exe -scriptargs "MaxwellDesign1 Setup1" -
  RunScriptAndExit C:\temp\test.py
  ```

  In Python, the command line parameter following **-scriptargs** is passed without modification as a single string in the ScriptArgument python variable.

  In VBscript, the command line parameter following **-scriptargs** is split into multiple strings and converted to a VBscript collection which is accessible via the **AnsoftScript.Arguments** collection. To access these arguments, for example:

  ```
  msgbox AnsoftScript.Arguments(0) // Returns MaxwellDesign1
  msgbox AnsoftScript.Arguments(1) // Returns Setup1
  ```

  In either case, **Design1** is taken into Maxwell as the first argument, and **Setup1** as the second argument. If you failed to use quotation marks, **Design1** would be taken as the first argument and **Setup1** would not be understood by Maxwell.

-Batchoptions

All options that are specified through **Tools** > **Options** go into the user-level registry. You can override these registry entries via the **-batchoptions** command line. These overrides apply only to the current Desktop session. The registry setting overrides may be specified on the command line, or may be in a file with the file pathname specified on the command line. Batch jobs can be submitted from the command line or through Electronics Desktop's job submission window.

Large Scale DSO offers two new batchoptions related to the redistribution ability.

- `LargeScaleDSO/VarRedistribution`, where 0 disables redistribution (default), and 1 enables it.
- `LargeScaleDSO/RedistributionLimit`, is a positive integer specifying the minimum estimated remaining time (in minutes) for variations to redistribute to another task. The default is 3.

> **Note:**
>
> **-batchoptions** is only valid for batch jobs. It is ignored if you have not specified **-BatchSolve**, **-BatchSave**, or **-BatchExtract**.

-Batchoptions Examples

The batchoption **CreateStartingMesh** is available for all 3D products (HFSS, HFSS-3DLayout, Q3D, Q2D, Maxwell3D and Maxwell2D). When this option is set, only the initial mesh and manual mesh operation making portion of the setup are completed for the batch solve.

This example enables CreateStartingMesh for HFSS, and runs a batch solve on the specified project:

```
ansysedt -batchoptions "'HFSS/CreateStartingMesh'=1" -
batchsolve "D:\projects\MyProject.aedt"
```

See: Additional Examples of -Batchoptions Use.

Export Options Files

The **Tools** > **Options** > **Export Options Files** command writes XML files containing the options settings at all levels to the specified directory. This feature is intended to make it easier for different users to use Ansys Electromagnetics Suite 2023 R1 installed on shared directories or network drives. See: Example Uses for Export Options Features.

Related Topics

Running a Script

Examples and Further Explanations of -batchoptions Use

This section provides examples and further explanations of -batchoptions.

- Example with registry settings specified on the command line
- Example with registry settings specified in a file
- When to use the -batchoptions Desktop command line option

The following examples use general Desktop and Maxwell-specific settings. This feature is available for all desktop products.

- The registry path separator is the slash (/) character.
- Each complete registry key (that is, a registry path and option name) is enclosed in single quotes.
- Registry string values are enclosed in single quotes.
- After the -batchoptions switch, the set of registry keys and values that follows it must be enclosed in double quotes. However, if a batchoptions file is referenced (instead of listing the options directly on the command line), the double quotes are not used around the filename.
- Backslashes in registry key string values must be escaped with another backslash (\\ ), since a backslash by itself is an escape code within strings.

Example with registry settings specified on the command line

```
ansysedt.exe -batchsolve -batchoptions
"'Desktop/Settings/ProjectOptions/NumberOfProcessors'=4
'Maxwell3D/NumCoresPerDistributedTask'=2    (* This option is not
currently available from the Add Batchoption dialog box.)
'Desktop/ProjectDirectory'='C:\\projects\test'" projectname.aedt
```

This command line overrides registry values of the NumberOfProcessors (Desktop/Settings/ProjectOptions key) and ProjectDirectory (Desktop key) options.

> **Note:**
>
> - Multiple registry settings may appear in a single -batchoptions value, separated by whitespace.
> - The -batchoptions value must be enclosed in double quotes if it contains any whitespace.

Example with registry settings specified in a file

```
maxwell.exe -batchsolve -batchoptions filename projectname.aedt
```

where the referenced file, <*filename*> contains:

```
$begin 'Config'
  'Desktop/Settings/ProjectOptions/NumberOfProcessors'=4
  'Maxwell3D/NumCoresPerDistributedTask'=2
  'Desktop/ProjectDirectory'='C:/projects/test'
$end 'Config'
```

This command overrides the registry values of the NumberOfProcessors (Desktop/Settings/ProjectOptions key), UpdateReportsOnSolve (Desktop/Settings/ProjectOptions/Maxwell3D key), and ProjectDirectory (Desktop key) options. These overrides apply only to the current Electronics Desktop session.

> **Note:**
>
> - The -batchoptions ***<filename>*** value must be enclosed in double quotes if it contains whitespace
> - The **$begin 'Config'** and **$end 'Config'** lines are required

For additional options you can override for the command line with -batchoptions, see

- For -batchoptions Use: Project Directory and Lib Paths
- For -batchoptions Use: TempDirectory
- For -batchoptions Use: Various Desktop Settings
- For -batchoptions Use: Maxwell2D and Maxwell3D Options

When to use the -batchoptions Desktop Command Line Option

You can set analysis parameters for batch mode jobs using the graphical user interface (GUI). For example, you can set all options for a batch job using the **Add Batch Option** dialog box, which is accessed through the **Submit Job To** dialog box. These parameter settings include the following solver options (several examples, not a complete list):

- HPCLicenseType
- tempdirectory
- Desktop/AutoExtractReports
- Desktop/Settings/ProjectOptions/MumberOfProcessors
- <*design_type*>/DefaultProcessPriority
- <*design_type*>/EnableGPU – Applicable to Maxwell 3D
- <*design_type*>/MPIVendor (either “Intel” or “Microsoft”) – Applicable to Maxwell 2D and Maxwell 3D

For graphical analyses that do **not** use batch mode, you specify the analysis parameters using the GUI (**Tools** > **Options** > **General Options** or **HPC and Analysis Options**). These settings are written to the registry when you exit the Electronics Desktop program. The settings are read from the registry when the application is started. Therefore, when you start the Electronics Desktop application, all settings retain the values from the previous session of the same user on the same machine. If there was not a previous session of the same user on the same machine, then the values are obtained from other registry configuration files or from a default value.

When running a batch analysis, any setting that is not specified using the **-batchoptions** command line option is taken from the registry. This value is typically the setting from the last session of the same user on the same machine. However, the **-batchoptions** command line option allows you to override the parameter with values specified on the command line or in a batchoptions file. The values specified using the **-batchoptions** command line option only apply to the batch job, and do not affect the parameter values stored in the registry.

If important **-batchoptions** values are not specified when running a batch job, the paramaters could be affected by an interactive session running on the same host by the same user. Parameter changes can occur if the user sets an option in the GUI and then exits the program, or if another process that accesses the registry exits. To be sure of the desired batch job outcome, avoid changing options in concurrent interactive sessions or include the desired -batchoptions in the command line.

For -batchoptions Use: Project Directory and Lib Paths

The PersonalLib, syslib and userlib settings are a little different from other settings. If the final directory name is different from what is expected, then PersonalLib, syslib or userlib is appended as a final directory. In addition, these settings may come from a different registry value if the registry values shown above are not set.

Registry Key	Default Value	Units or Values	Description
Desktop/ProjectDirectory	Ansoft subdirectory of user's HOME directory or "Documents" directory	Directory pathname	Directory where new projects are created
Desktop/PersonalLib	PersonalLib subdirectory of user's HOME directory or	Directory pathname	Directory PersonalLib is appended if final

Registry Key	Default Value	Units or Values	Description
	"Documents" directory		directory is not PersonalLib
Desktop/syslib	syslib subdirectory of installation directory	Directory pathname	Directory syslib is appended if final directory is not syslib
Desktop/userlib	userlib subdirectory of installation directory	Directory pathname	Directory userlib is appended if final directory is not userlib

Related Topics

For -batchoptions Use: TempDirectory
For -batchoptions Use: Various Desktop Settings
For -batchoptions Use: Maxwell2D and Maxwell3D Options with Paths
Running Maxwell from a Command Line

For -batchoptions Use: TempDirectory.

Registry Key	Default Value	Units or Values	Description
TempDirectory	Set by installer	-	Directory for temporary files

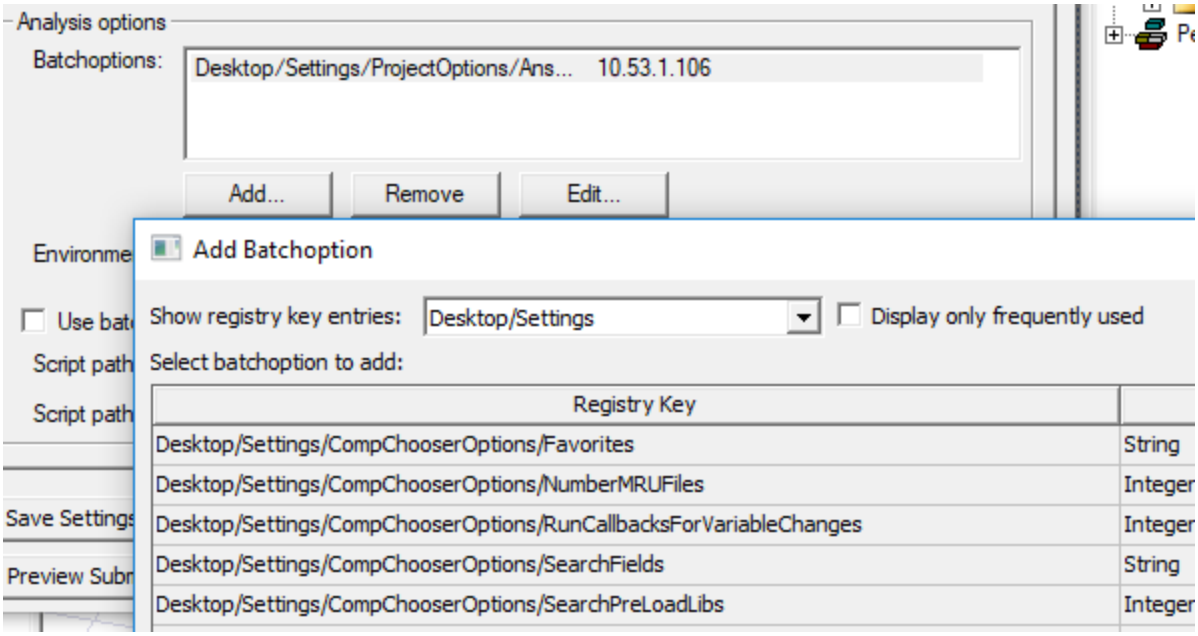
Related Topics

For -batchoptions Use: Project Directory and Lib Paths
For -batchoptions Use: Various Desktop Settings
For -batchoptions Use: Maxwell2D and Maxwell3D Options
Running Maxwell from a Command Line

For -batchoptions Use: Various Desktop Settings

Note that most of these options only affect the GUI. To view these options from the *Submit Job To* window:

1. In the *Analysis options* area of the *Submit Job To* window, click **Add...**.
2. In the *Add Batchoption* window that appears, select **Desktop/Settings** from the **Show registry key entries** drop-down menu.



Registry Key	**Default Value**	**Units or Values**	**Description**
Desktop/Settings/ProjectOptions/AnimationMemory	200	Megabytes (MB)	Stop animations when available memory falls below this value
Desktop/Settings/ProjectOptions/AnsoftCOMPreferredIPAddress (see note below table)	“” (empty string)	IP address (as a string)	IP address used to connect from COM engines to ansysedt.exe

Registry Key	Default Value	Units or Values	Description
Desktop/Settings/ProjectOptions/AnsysEMPreferredSubnetAddress (see note below table)	“” (empty string)	IP address (as a string)	Subnet used to connect COM engines to ansysedt.exe – allowed formats are: • IPv4 network prefix in CIDR notation Example: 123.123.123.0/24 • IPv4 network prefix with /subnet mask appended Example: 123.123.123.0/255.255.255.0
Desktop/Settings/ProjectOptions/AutoSaveInterval	10	edits	Number of edits to allow between autosaves
Desktop/Settings/ProjectOptions/AutoShowMessageWindow	1 (true)	0 (false) or 1 (true)	Show message window on new messages
Desktop/Settings/ProjectOptions/AutoShowProgressWindow	0 (false)	0 (false) or 1 (true)	Show progress window when starting a simulation
Desktop/Settings/ProjectOptions/DiskLimitForAbort	0	Megabytes (MB)	A warning is issued when available disk space falls below this value
Desktop/Settings/ProjectOptions/DoAutoSave	1 (true)	0 (false) or 1 (true)	Enables autosaves if true
Desktop/Settings/ProjectOptions/DrawStateIconsInProjectTree3	1 (true)	0 (false) or 1 (true)	Change icon when selection does not match active window
Desktop/Settings/ProjectOptions/ExpandMessageTreeOnInsert	1 (true)	0 (false) or 1 (true)	Ensure that new messages are visible in the message window tree

Registry Key	Default Value	Units or Values	Description
Desktop/Settings/ProjectOptions/ExpandOnInsert	0 (false)	0 (false) or 1 (true)	Expand project tree on insert
Desktop/Settings/ProjectOptions/HighlightActiveContextInProjectTree2	1 (true)	0 (false) or 1 (true)	Emphasize active command context (menu and toolbars)
Desktop/Settings/ProjectOptions/SavePreviewImagesInProjectFile	1 (true)	0 (false) or 1 (true)	Save preview images in project file
Desktop/Settings/ProjectOptions/UpdateReportOnFileOpen	0 (false)	0 (false) or 1 (true)	Update reports on file open

Note:

The preferredIP address and preferred subnet address settings are mutually exclusive. If both are specified to be non-empty strings, then the preferred IP address takes precedence, and the preferred subnet address is ignored. This feature is typically used for cluster environments using batch solves. The setting can be made via batchoptions but can also be done via UpdateRegistry.

Related Topics

For -batchoptions Use: Project Directory and Lib Paths

For -batchoptions Use: TempDirectory

For -batchoptions Use: Maxwell2D and Maxwell3D Options

Running Maxwell from a Command Line

For -batchoptions Use: Maxwell 2D and Maxwell 3D Options

Here are the Maxwell Options with the complete registry keys (that is, the registry paths and option names). You can access a help window with information covering most available design-

type-specific batchoptions using the **-batchoptionhelp** switch when launching Electronics Desktop from a command line:

Registry Key	Default Value	Units or Values	Description
Maxwell 2D (or 3D)/DesiredRAMLimitInGB	0 (No Limit)	GB	Preferred memory use in GB. Type: Floating Point, Min: 0, Max: 1000000.
Maxwell 2D (or 3D)/MaxRAMLimitInGB	0 (No Limit)	GB	This setting specifies the absolute maximum memory usage in GB. Type: Floating Point, Min: 0, Max: 1000000.
Maxwell 2D (or 3D)/HPCLicenseType	"pack"	"pool" or "pack"	For Maxwell 2D Designs: "pool" = multiprocessing enabled by HPC licensing; "pack" = multiprocessing enabled by HPC Pack licensing
Maxwell 2D(or 3D)/DefaultProcessPriority	Normal	Allowed values: "Critical", "Above Normal", "Normal", "Below Normal", "Idle"	Specifies the default priority of distributed tasks.
Maxwell 2D(or 3D)/HPCLicenseType	“pack”	“pool” or “pack” or "none"	Specifies if HPC license is used and which type.
Maxwell 2D(or 3D)/MPIVendor		“Microsoft”, “Intel”.	Which MPI libraries to use for remote communication. Type: String,
Maxwell 2D(or 3D)/MPIVersion	"Default"	"Default", "2018", "2021"	This setting specifies the MPI version. If not specified, then the default version for the MPI vendor will be used. The valid values depend on the MPI vendor. Allowed values: "Default", "2018", "2021"
Maxwell 2D(or 3D)/RemoteSpawnCommand			This setting specifies the command that will be used to launch distributed solver jobs. The value Scheduler is only

Registry Key	Default Value	Units or Values	Description
			valid for jobs running under an LSF or SGE scheduler, and only if the MPI Vendor is Intel. Type: String, Allowed Values: "RSH", "SSH", "Scheduler".
Maxwell 2D(or 3D)/DistribLicense		"DSO", "Solver"	This setting specifies which license feature to use for distributed variations. Type: String
Maxwell 2D(or 3D)/NumCoresPerDistributedTask		Min: 1, Max: 1000	This setting specifies the number of cores that will be used per distributed task when using a machine list provided by the scheduler. Type, Integer.
Maxwell 2D(or 3D/SolveAdaptiveOnly		0 (False), 1 (True).	If true, only the adaptive portion of the setup will be solved. Applies only if a single setup is selected.
Maxwell 2D(or 3D)/CreateStartingMesh		0 (False), 1 (True)	If true, the starting mesh (with mesh operations if applicable) will be created, but none of the setup will be solved. Type: Integer
Maxwell 2D(or 3D)/RAMLimitPercent		Min: 1, Max: 99	This setting specifies the percent of total RAM to be used on each machine for solving. Please be sure to leave sufficient RAM for the operating system and applications. Type: Integer
Maxwell 2D(or 3D)/RAMLimitPerCoreInGB		Min: 0, Max: 10000	This setting specifies the maximum amount of RAM used for each core allocated by the scheduler in GB. This cannot be combined with RAM limit percent and is only valid when solving in a scheduler environment. Type: Floating Point
Maxwell 2D(or 3D)TotalNumOfCores		Min: 1, Max: 1000	This setting specifies the total

Registry Key	Default Value	Units or Values	Description
			number of cores that will be used across all tasks when using machine list provided by the scheduler. Type: Integer
Maxwell 2D(or 3D)/SolveAdaptiveOnly	1	Allowed values: 0 (false), 1 (true)	If true, only the adaptive portion of the setup will be solved. Applies only if a single setup is selected.
Maxwell 2D(or 3D)/CreateStartingMesh	0	Allowed values: 0 (false), 1 (true)	If true, the starting mesh (with mesh operations if applicable) will be created, but none of the setup will be solved. Applies only if a single setup is selected.
Maxwell 2D/RAMLimitPercent		Min: 1, Max: 99	This setting specifies the percent of total RAM to be used on each machine for solving. Please be sure to leave sufficient RAM for the operating system and applications. Type: Integer
Maxwell 2D(or 3D)/RAMLimitPerCoreInGB		Min: 0, Max: 10000	This setting specifies the maximum amount of RAM used for each core allocated by the scheduler in GB. This cannot be combined with RAM limit percent and is only valid when solving in a scheduler environment. Type: Floating Point
Maxwell 2D(or 3D)/DefaultProcessPriority	Normal	"Critical", "Above Normal", "Normal", "Below Normal", "Idle"	Specifies the default priority of distributed tasks.
Maxwell 3D/EnableGPU	0	0 (False), 1 (True).	Allow GPU to be used for eddy current solves. GPU acceleration will be disabled when automatic setting is selected in Analysis

Registry Key	Default Value	Units or Values	Description
			Configuration. Type: Integer

Related Topics

For -batchoptions Use: Project Directory and Lib Paths

For -batchoptions Use: TempDirectory

For -batchoptions Use: Various Desktop Settings

Running Maxwell from a Command Line

Running from a Windows Remote Terminal

When running Maxwell from a remote terminal, there are some performance and behavior issues to consider. These issues are due to the interaction of bandwidth/opengldrivers/remote-terminal-protocol

- Showing axes when interactively drawing objects will slow the performance.
- Remote OpenGL performance will be slower in general. Graphics card and driver quality helps.
- All 3D windows will be closed when you switch from remote PC to a console or from a console to remote. This is to avoid display/opengl instability during the switch.
- Grid will not be turned off while viewing a plot from a remote desktop. The mouse over highlights on 2D plots may appear as not totally overlapping the line color or as thin dotted lines.

Related Topics

Remote Analysis

Modifying the Model View

Windows HPC Commands

HPC Integration allows you to submit jobs directly using Ansys Electromagnetics command line arguments for batchsolves. The supported HPC software is described in the Ansys Electromagnetics Installation Guide. Ansys Electromagnetics products must be accessible from the same directory on all machines. The Ansys Electromagnetics command line syntax is documented here. You must pass in a -distributed flag as part of the Ansys Electromagnetics command line arguments if you want to run a distributed simulation.

Before running a job you must select **Tools>JobManagement>Select Scheduler** and use the dialog to designate the head node of a cluster. You can then select**Tools>Job Mangement>Submit Job** to submit the batch commands for the job.

Related Topics

High Performance Computing (HPC) Integration

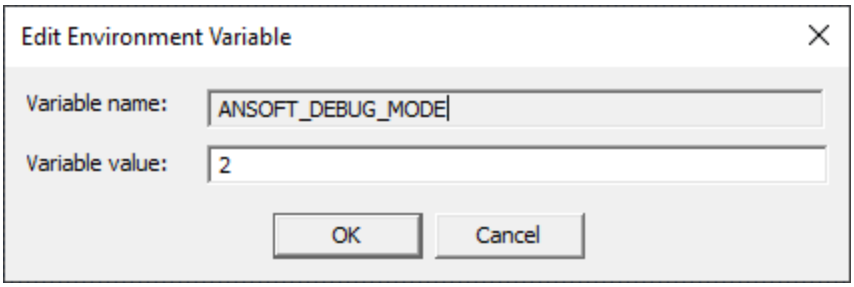
Debug Logging

Electronics Desktop provides error logging capabilities.

To begin logging errors:

1. Click **Tools** > **Debug Logging**.

   The **Edit Environment Variable** window appears, showing the ANSOFT_DEBUG_MODE variable.

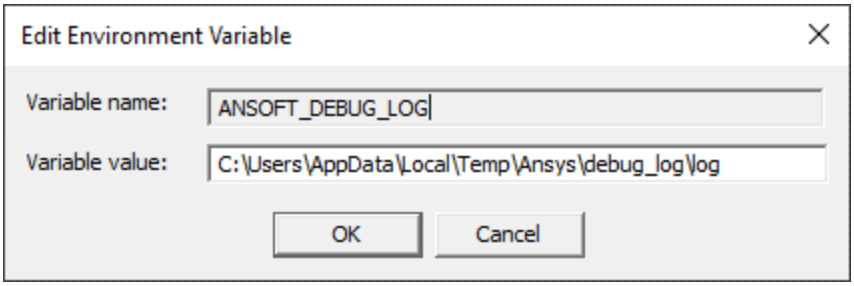


2. Enter a **Variable value**.

   The default value is 2, and should not be changed unless directed by technical support.

3. Click **OK**.

   The **Edit Environment Variable** window updates to display the ANSOFT_DEBUG_LOG variable.



4. Enter a folder path where log files (*.log) will be stored.
5. Click **OK**.

   Errors are logged to the specified folder until you disable logging.

To stop logging errors:

- Click **Tools** > **Stop Debug Logging**.

3 - Help Menu Options

The Help Menu displays different selections depending upon the type of design inserted to the active project (such as Maxwell, HFSS). The following describes the basic selections displayed on the Help Menu.

- **Maxwell Help** opens to the Maxwell help within the Electronics help system. You can also access PDF versions from within the help system.
- **Maxwell Scripting Help** opens to Maxwell's scripting help.
- **Maxwell Getting Started Guides** opens to a list of links to the Maxwell Getting Started Guides. These Getting Started Guides walk you through projects that demonstrate features of the product solvers.
- **Maxwell PDFs**provides access to PDFs for Maxwell, including the main help, scripting guide, and Getting Started Guides.
- **Maxwell Components** is enabled when you insert a Maxwell Circuit design into the active project, and opens the Maxwell Components help to a section that pertains to the component type.
- **Ansys Customer Support** opens a browser page to the Ansys Customer Portal. At the website you can learn more about Ansys products and services and log on to contact Ansys technical support staff.
- **What's New in this Release** opens a PDF that describes *What's New in Ansys Electronics Desktop* for the release.
- **Ansys Product Improvement Program** opens a dialog that describes the Product Improvement Program option.

Note	You can disable the Ansys Product Improvement Program for all users so that each user is not prompted to enable the Program when they first start Electronics Desktop. To do this after installing the software, run the following command as a user with permissions to modify the installed fileset: **UpdateRegistry.exe -set -ProductName ElectronicsDesktop2023.1 -RegistryKey Desktop/Settings/ProjectOptions/ProductImprovementOptStatus -RegistryLevel install -RegistryValue 1**

- **About Ansys Electronics Desktop** opens a dialog that displays the Ansys® Electromagnetics Suite release number and contains tabs that show information about the **Installed Components** and **Client License Settings**.

F1 Context-Sensitive Help

To access **F1** help from the Ansys Electronics Desktop user interface, do one of the following:

- To open a help topic about a menu option, hover the cursor over the item and press F1. You can also press **Shift**+**F1** on the item.
- To open a help topic about a dialog box, open the dialog box, and then press **F1**.

Related Topics

Scripting in Maxwell

Conventions Used in the Help

Please take a moment to review how instructions and other useful information are presented in this documentation.

- Procedures are presented as numbered lists. A single bullet indicates that the procedure has only one step.
- Bold type is used for the following:
  - Keyboard entries that should be typed in their entirety exactly as shown. For example, "**copy file1**" means you must type the word **copy**, then type a space, and then type **file1**.
  - On-screen prompts and messages, names of options and text boxes, and menu commands. Menu commands are often separated by greater than signs (>). For example, "click **HFSS** > **Excitations** > **Assign** > **Wave Port**."
  - Labeled keys on the computer keyboard. For example, "Press **Enter**" means to press the key labeled **Enter**.
- Italic type is used for the following:
  - Emphasis
  - The titles of publications
  - Keyboard entries when a name or a variable must be typed in place of the words in italics. For example, "**copy** *filename*" means you must type the word **copy**, then type a space, and then type the name of the file.
- The plus sign (+) is used between keyboard keys to indicate that you should press the keys at the same time. For example, "Press Shift+F1" means to press the **Shift** key and, while holding it down, press the **F1** key also. You should always depress the modifier key or keys first (for example, Shift, Ctrl, Alt, or Ctrl+Shift), continue to hold it/them down, and then press the last key in the instruction.

Accessing Commands: *Ribbons, menu bars,* and *shortcut menus* are three methods that can be used to see what commands are available in the application.

- The *Ribbon* occupies the rectangular area at the top of the application window and contains multiple tabs. Each tab has relevant commands that are organized, grouped, and labeled. An example of a typical user interaction is as follows:

  "Click **Draw** > **Line**"

  This instruction means that you should click the **Line** command on the **Draw** ribbon tab. An image of the command icon, or a partial view of the ribbon, is often included with the instruction.

- The *menu bar* (located above the ribbon) is a group of the main commands of an application arranged by category such File, Edit, View, Project, etc. An example of a typical user interaction is as follows:

  "On the **File** menu, click the **Open Examples** command" means you can click the **File** menu and then click **Open Examples** to launch the dialog box.
- Another alternative is to use the *shortcut menu* that appears when you click the right-mouse button. An example of a typical user interaction is as follows:

  “Right-click and select **Assign Excitation> Wave Port**” means when you click the right-mouse button with an object face selected, you can execute the excitation commands from the shortcut menu (and the corresponding sub-menus).

Getting Help: Ansys Technical Support

For information about Ansys Technical Support, go to the Ansys corporate Support website, http://www.ansys.com/Support. You can also contact your Ansys account manager in order to obtain this information.

All Ansys software files are ASCII text and can be sent conveniently by e-mail. When reporting difficulties, it is extremely helpful to include very specific information about what steps were taken or what stages the simulation reached, including software files as applicable. This allows more rapid and effective debugging.

Help Menu

To access help from the Help menu, click **Help** and select from the menu:

- **[product name] Help** - opens the contents of the help. This help includes the help for the product and its *Getting Started Guides*.
- **[product name] Scripting Help** - opens the contents of the *Scripting Guide*.
- **[product name] Getting Started Guides** - opens a topic that contains links to Getting Started Guides in the help system.

Context-Sensitive Help

To access help from the user interface, press **F1**. The help specific to the active product (design type) opens.

You can press **F1** while the cursor is pointing at a menu command or while a particular dialog box or dialog box tab is open. In this case, the help page associated with the command or open dialog box is displayed automatically.

Finding Information in the Help

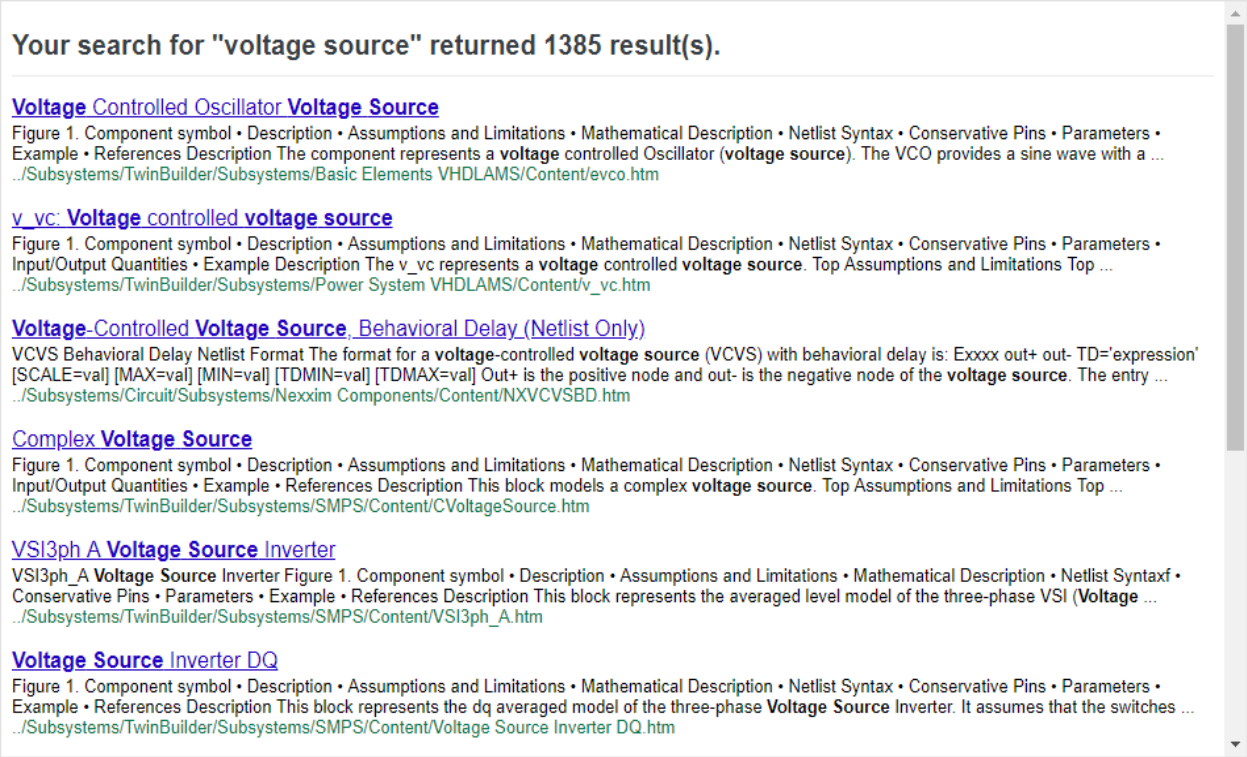
The help system provides four ways to find information and navigate quickly:

- A hierarchical table of contents - To browse through the table of contents, click the **Contents** tab. The table of contents is an expandable list of important topics. You can expand or collapse the list, and then jump to detailed information for a topic by clicking the topic title.

- A full text search - you can type text, and search the entire help. Items are listed according to rank in discussing the search text. For more information, see Using the Search Function in the Help.

Using the Search Function in the Help

When you enter words or strings to search for in the help, the search engine, by default, lists all topics in which any of the words occur. For example, if you enter “voltage source” without the quotation marks, the results show all topics that contain “voltage” or “source.”

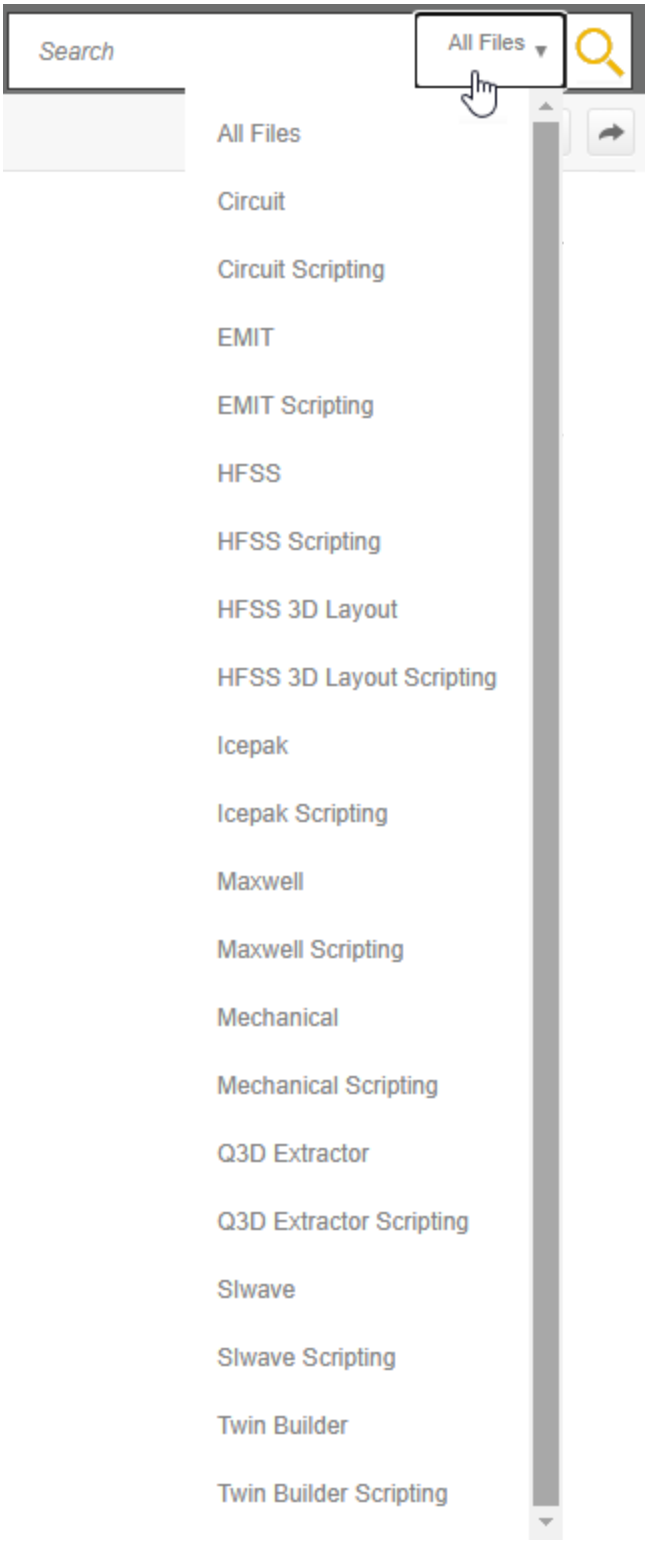


This method probably provides more hits than you want. The Search function in the help provides several methods for making searches more specific.

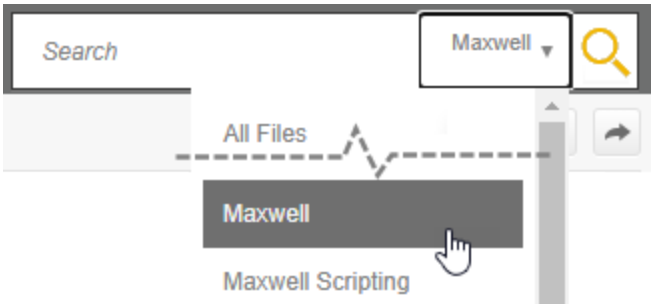
Performing a Basic Search

1. Type the words or string you are searching for in the search box.
    - If you are searching within the full Electronics help system, the search box includes a drop-down filter for specifying a product but not including its scripting guide, specifying a product's scripting guide, or searching across all products. When you

change the filter, the results dynamically reflect the selected filter.



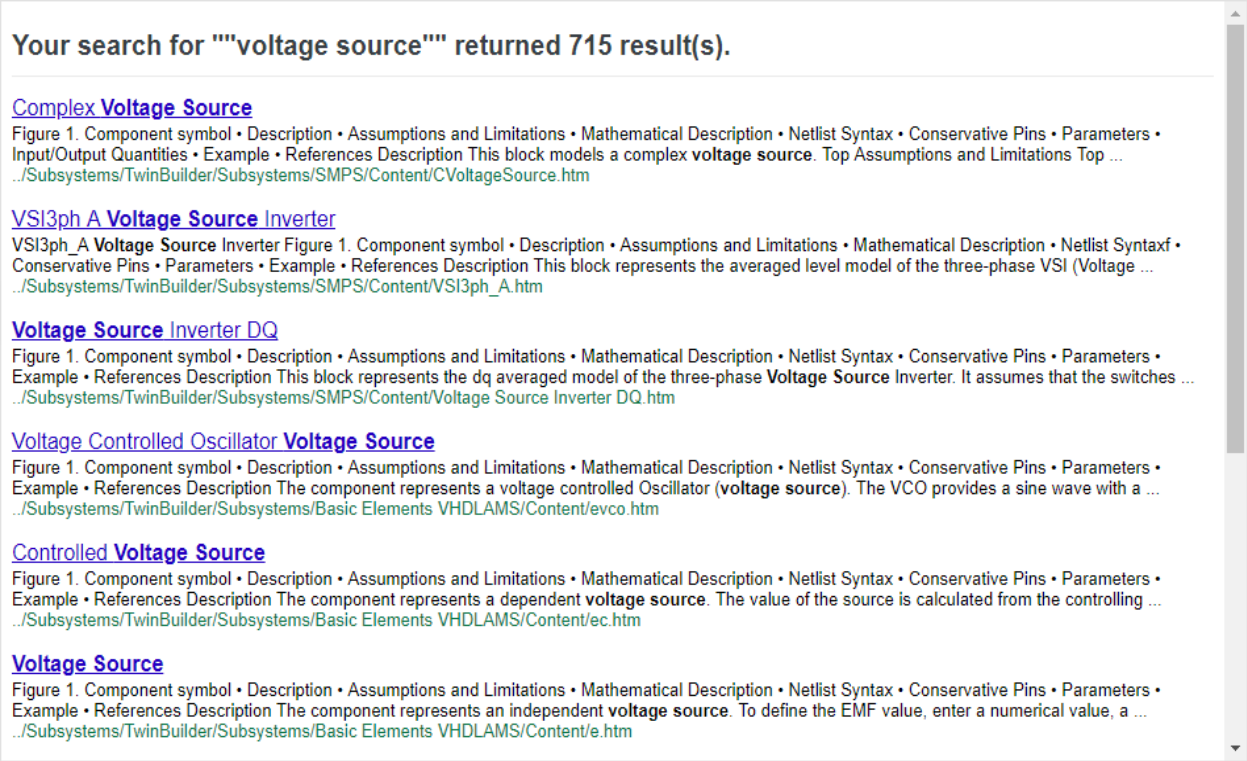
- If you used F1 from Maxwell, you see a search box that permits searches for all files in Maxwell (for example, *All Files*), all files except the Maxwell scripting guide (for example, *Maxwell*), or just files in the *Maxwell* scripting guide (for example, *Maxwell Scripting*).



2. Click on the topic you want in the list results.
   - If you want to view a different topic, press your browser's back button to return to the list results.
   - To turn off highlighting on the page you are viewing, click the Remove Highlights icon

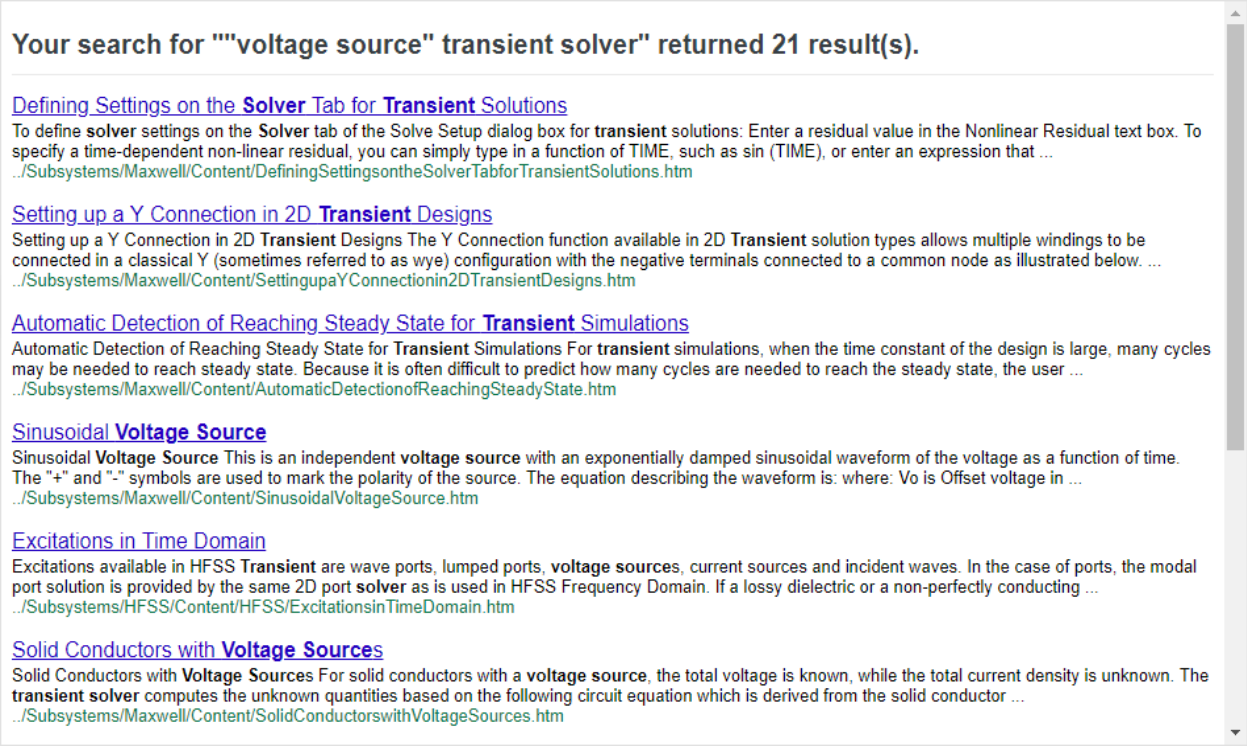
Searching with Quotation Marks

If you enter “voltage source” with quotation marks, the results show all topics that include the phrase.



As you can see, this returns far fewer results than the basic search.

To further limit the results, you can enter additional words, such as: “voltage source” transient solver



Note:

- Searches are not case sensitive, so you can type your search in uppercase or lowercase characters.
- You may search for any combination of letters (a-z) and numbers (0-9).
- Punctuation marks (period, colon, semicolon, comma, hyphen) are ignored during a search.
- When searching for a file name with an extension, group the entire string in quotation marks (for example, "filename.ext").

Using Boolean Operators

You can also use boolean operators to affect the number of topics listed.

Operator (s)	Usage	Example(s)
AND + &	Lists all topics that contain all of the terms.	Net AND Selection Net + Selection Net & Selection
OR |	Lists all topics that contain any of the terms.	Net OR Selection Net | Selection

Operator (s)	Usage	Example(s)
NEAR	Lists all topics that contain the terms near the other terms.	Net NEAR Selection
NOT ! ^	Lists all topics that contain the first term but not the second.	Net NOT Selection Net ! Selection Net ^ Selection

Use parentheses to group terms and operators. For example:

- solver AND (Circuit) NOT HFSS NEAR dynamic
- solver AND (HFSS OR Circuit) NOT (Q3D OR 2d) NEAR dynamic
- “dynamic link” ! (HFSS | circuit)

> **Important:**
>
> Because the characters +, &, |, !, and ^ are used as operators, you cannot search for them in the help. Doing so will result in an error.

The Ansys Product Improvement Program

This product is covered by the Ansys Product Improvement Program, which enables ANSYS, Inc., to collect and analyze anonymous usage data reported by our software without affecting your work or product performance. Analyzing product usage data helps us to understand customer usage trends and patterns, interests, and quality or performance issues. The data enable us to develop or enhance product features that better address your needs.

How to Participate

The program is voluntary. To participate, select **Yes** when the Product Improvement Program dialog appears. Only then will collection of data for this product begin.

How the Program Works

After you agree to participate, the product collects anonymous usage data during each session. When you end the session, the collected data is sent to a secure server accessible only to authorized Ansys employees. After Ansys receives the data, various statistical measures such as distributions, counts, means, medians, modes, etc., are used to understand and analyze the data.

Data We Collect

For all products that offer the Ansys Product Improvement Program, we only collect anonymous data such as session statistics, hardware information, types of loading, solution types, solution statistics, and similar data. The specific data collected varies from product to product.

For Ansys Electronics, we collect the following information:

- Application

    - Build information
    - System information
        - Country
        - Country code
        - CPU architecture
        - CPU brand
        - CPU identifier
        - Graphics card
        - Operating system
        - Operating system version
        - Processor count
        - Time zone
        - Total RAM value
- Session
    - Workbench session
    - Total CPU time
    - Execution mode
    - Start method
    - Number of processes
    - Number of compute nodes (HPC)
    - Session begin
    - Session end
- Mesh
    - Number of nodes
    - Number of elements
    - Number of zones
    - Number of faces

Data We Do Not Collect

The Product Improvement Program does not collect any information that can identify you personally, your company, or your intellectual property. This includes but is not limited to names, addresses, file names, part names, geometry- or design-specific inputs, material property values, etc. We make no record of where we collect data from.

Opting Out of the Program

You may stop your participation in the program any time you wish. To do so, select **Ansys Product Improvement Program** from the Help menu. A dialog appears and asks if you want to continue participating in the program. Select **No** and then click **OK**. Data will no longer be collected or sent.

The ANSYS, Inc., Privacy Policy

All Ansys products are covered by the ANSYS, Inc., Privacy Policy, which you can read here.

Frequently Asked Questions

1. *Am I required to participate in this program?*

   No, your participation is voluntary. We encourage you to participate, however, as it helps us create products that will better meet your future needs.
2. *Am I automatically enrolled in this program?*

   No. You are not enrolled unless you explicitly agree to participate.
3. *Does participating in this program put my intellectual property at risk of being collected or discovered by ANSYS?*

   No. We do not collect any project-specific, company-specific, or model-specific information.
4. *Can I stop participating even after I agree to participate?*

   Yes, you can stop participating at any time. To do so, select **Ansys Product Improvement Program** from the Help menu. A dialog appears and asks if you want to continue participating in the program. Select **No** and then click **OK**. Data will no longer be collected or sent.
5. *Will participation in the program slow the performance of the product?*

   No, the data collection does not affect the product performance in any significant way. The amount of data collected is very small.
6. *How frequently is data collected and sent to Ansys servers?*

   The data is collected during each use session of the product. The collected data is sent to a secure server once per session, when you exit the product.
7. *Is this program available in all Ansys products?*

   Not at this time, although we are adding it to more of our products at each release. The program is available in a product only if this Ansys Product Improvement Program description appears in the product documentation, as it does here for this product.
8. *If I enroll in the program for this product, am I automatically enrolled in the program for the other Ansys products I use on the same machine?*

   Yes. Your enrollment choice applies to all Ansys products you use on the same machine. Similarly, if you end your enrollment in the program for one product, you end your enrollment for all Ansys products on that machine.
9. *How is enrollment in the Product Improvement Program determined if I use Ansys products in a cluster?*

   In a cluster configuration, the Product Improvement Program enrollment is determined by the host machine setting.

Getting Help from Ansys Technical Support

For information about Ansys Technical Support, go to the Ansys corporate Support website, www.ansys.com/Support. You can also contact your Ansys account manager to obtain this information.

E-mail can work well for technical support. All Maxwell software files are ASCII text and can be sent conveniently by e-mail. When reporting difficulties, it is extremely helpful to include very specific information about what steps were taken or what stages the simulation reached. This allows more rapid and effective debugging.

4 - Working with Maxwell Projects and Designs

A Maxwell project is a folder that includes one or more Maxwell models, or *designs*. Each design ultimately includes a geometric model, its boundary conditions and material assignments, and field solution and post-processing information.

A new project called *Projectn* is automatically created when the software is launched, where *n* is a number. If you select the **Insert a design of type** radio button on the Desktop Configuration options panel, a design of the specified type is automatically created for the new project. You can also open a new project by clicking **File>New**. In general, use the **File** menu commands to manage projects. If you move or change the names of files without using these commands, the software may not be able to find information necessary to solve the model.

Maxwell Files

When you create an Ansys Electronics Desktop project, it is given an **.aedt** file extension and stored in the directory you specify. Any files related to that project are also stored in that directory.

Some common file and folder types are listed below:

.aedt	Maxwell project.
project_name. **aedtresults**	Maxwell folder containing results data for a project.
design_name. **results**	Maxwell folder containing results data for a design. This folder is stored in the *project_name.* **aedtresults folder.**
design_name. **asol**	Maxwell results data for a design. This file's contents may be empty if a solution is unavailable. This file is stored in the *project_name.* **aedtresults** folder.
.cls	Ansoft legacy EM projects.

Creating Projects

1. Click **File>New**  .

   A new project is listed in the project tree. It is named Project*n* by default, where *n* is the order in which the project was added to the current session.

   Project definitions, such as material assignments, are stored under the project name in the project tree.

2. Click **Project>Insert Maxwell (2d or 3D) Design**.

You specify the name of the project when you save it using the **File>Save** or **File>Save As** commands.

Note	To insert an RMxprt design, click **Project>Insert RMxprt Design**.

Creating New 2D Designs from 3D Designs

The Maxwell system contains both 2D and 3D projects. Often, you may want to take a cross-section of a 3D model to perform a 2D analysis. The following cases represent the scenarios available for automatically converting models.

- 3D model to 2D XY model
- 3D model to 2D RZ model

You must keep the following points in mind before you convert any project:

- A 2D design is created in the same project.
- All 3D geometry is used, you cannot select which geometry to include in the conversion.
- If there are any sheet objects in 3D, only those sheets that lie in the same plane as the section plane are converted.
- 1D objects are not converted from 3D to 2D.

Related Topics

Converting a 3D Model to 2D

Converting a 3D Model to 2D

To create a 2D design from an existing 3D design:

1. Click **Maxwell3D>Create 2D Design**.
   The **Create 2D Design** dialog box appears.
2. Select the **Coordinate System** to be used for creating sections in the 3D model.
3. Choose the **Section Plane** within the chosen coordinate system. You can choose XY, YZ, or ZX.
4. Select the **Geometry Mode**, to use for the target 2D design. You can choose either XY or RZ.
5. Click **OK** to create the 2D design.

The modeler creates a new design within the current project and generates the 2D model within the new design. All material properties and material assignments are copied to the new design.

Related Topics

Creating a Cross-Section

Creating New 3D Designs from 2D Designs

The Maxwell system contains both 2D and 3D projects. Often, you may want to sweep or rotate a 2D model to create a full 3D design. The following cases represent the scenarios available for automatically converting models from 2D to 3D.

- 2D XY model to 3D model
- 2D RZ model to 3D model

You must keep the following points in mind before you convert any project:

- A 3D design is created in the same project.
- All 2D geometry is used, you cannot select which geometry to include in the conversion.
- 1D objects are converted from 2D into 3D sheet objects.

Related Topics

Converting 2D XY Model to 3D
Converting 2D RZ Model to 3D

Converting 2D XY Model to 3D

To create a 3D design from an existing 2D XY design:

1. Click **Maxwell2D>Create 3D Design**.
   The **Sweep Along Vector** dialog box appears.
2. Enter a value in the **Length along z-axis** box to obtain a swept length of the model.
3. Choose the desired **Dimension** of the sweep distance entered from the pull-down menu.
4. Click **OK** to create the 3D design.

The modeler creates a new design within the current project and generates the 3D model within the new design. All material properties and material assignments are copied to the new design.

Related Topics

Sweeping Along a Vector

Converting 2D RZ Model to 3D

To create a 3D design from an existing 2D RZ design:

1. Click **Maxwell2D>Create 3D Design**.
   The **Sweep Around Z Axis** dialog box appears.
2. Enter the **Angle of Sweep** to obtain a full 360° 3D-model or a wedge model to a selectable degree.
3. Enter the desired **Draft Angle** and **Draft Type**.

Note	Entering a draft angle for a model where objects touch the z axis results in a error

	due to an invalid object being requested. Draft angles may only be used when no objects are touching the z axis.

4. Enter the **Number of Segments** to use in creating the 3D model. If you do not change the default value (=0), the project will be created with true surfaces.
5. Click **OK** to create the 3D design.

The modeler creates a new design within the current project and generates the 3D model within the new design. All material properties and material assignments are copied to the new design.

Related Topics

Sweeping Around an Axis

Opening Projects

Open a previously saved project using the **File>Open** command.

1. Click **File**>**Open**.

   The **Open** dialog box appears.
2. Use the file browser to find the Maxwell project file.

   By default, Maxwell files are displayed. To see other files, change the file filter in the **Files of type** pull-down list.
3. Select the file you want to open.
4. Click **OK**.

   The project information appears in the project tree.

   If you open another project without editing the automatically-created project, Maxwell removes the automatically-created project.

You can also open a saved project by:

- Dragging a Maxwell project file icon to the Maxwell icon.
- Dragging a Maxwell project file icon to the Maxwell desktop.
- Double-clicking on a Maxwell project file icon.

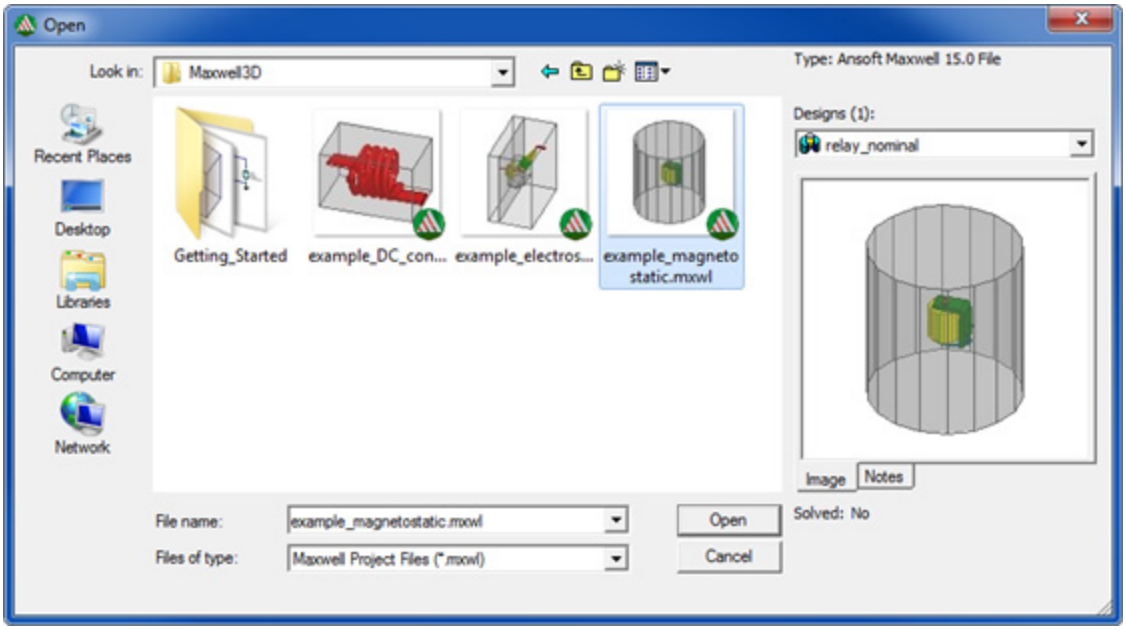
Related Topics

Opening Legacy Maxwell Projects

Opening Example Projects

Opening Example Projects

You can directly access and open example projects included with product install by using **File>Open Examples**. This displays a browser open to the Examples folder in the product install. You can select from various display styles for the folders and projects.



The help contains additional descriptions of these projects.

Related Topics

Example Projects

Opening Recent Projects

To open a project you recently saved in Maxwell:

- Click the name of the project file at the bottom of the **File** menu.

Note	If you open another project without editing the automatically-created project, Maxwell removes the automatically-created project.

Opening Legacy Maxwell Projects

Legacy Project Translation Considerations

When you open a legacy project, virtually all of the project's pre-processing data is translated. Solution results and Optimetrics setup data are unavailable; however, the nominal model created for Optimetrics is translated.

Objects that were considered non-model in the legacy project (**Model Object** check box was cleared in the Properties dialog) are translated but remain non-model objects in the translated project.

Following are additional notes about the translation of legacy project information:

Model Geometry	• The translated geometry's construction history is unavailable; therefore, the original object properties you defined cannot be modified in the **Properties** window. However, you can modify the geometry using Maxwell's modeling features. • For units unavailable in the older version, such as yards, the nearest available units are used; the model will be scaled slightly to fit the new units..
Excitations and Boundaries	• Boundaries assigned to named interface selections or rectangle selections are not translated. • For a boundary assigned to the intersection of two faces, Maxwell will create a new 2D sheet object from the intersecting area and assign the boundary to that object.
Materials	• Functions defined in legacy projects become project variables; therefore, functional material properties and functional material assignment angles are *not* translated. • Perfect conductors become regular materials with conductivity values of 1E30. Maxwell uses a conductivity threshold that treats all conductors over the threshold as perfect conductors. • Object coordinate systems are created for objects assigned anisotropic materials in legacy projects. The coordinate system is defined at the same origin as the global coordinate system, with the same orientation defined when the anisotropic material was assigned to the object in the legacy project. • Nonlinear materials from legacy projects that have magnetic saturation values greater than zero are treated as ferrite materials. Their properties are not modified. • When translating a legacy project, Maxwell does not import any variables that depend on intrinsic variables. For material properties depending on intrinsic variables, the variable reference is replaced with its expanded expression.
Mesh	• Mesh refinement operations performed on arbitrary boxes in legacy projects are ignored. • Area-based and volume-based mesh operations are translated as length-based mesh operations by taking their square roots and cube roots, respectively.
Optimetrics	• Setup information, including design variables, is not supported; however, the nominal model can be translated. • Parameterizing a translated model is limited because geometry construction history is unavailable.
Solution Setup	• The design's initial mesh is used for the solution. Meshes are not translated.
Solutions	• Solution data is not translated; therefore, you must solve legacy projects again.

Saving Projects

Use the **File>Save As** command to do the following:

- Save a new project.
- Save the active project with a different name or in a different location.

Use the **File>Save** command to save the active project.

Maxwell has a "Save before solving" setting located in the **Tools>Options>Maxwell** *(3D, or 2D)* **Options** or **Tools/Options/RMxprt Options** menu. By default this is on. However, for efficiency reasons, the project is only saved if it has been modified since its last save.

A prompt appears when you attempt to save a previously versioned file. If you agree to the prompt, the file is upgraded to the Maxwell version in which you are running the software. In this case the file may no longer be compatible with previous versions. If you do not agree to the prompt, the file is not saved, so the file retains the previous compatibility.

If you have a simulation running, you see a warning that if you continue, Maxwell will abort the simulation. If you OK the warning, Maxwell aborts the simulation and saves the project.

Path Name Length Issues for Windows

For most Windows programs, the current directory pathname length is limited to 259 characters on startup. Thus, the pathname length of the directory containing installed programs should be limited to no more than 259 characters because double-clicking on an application in Windows Explorer will set the working directory to the directory containing the application when the application is started.

Win32 Long Paths Not Enabled

If win32 long paths are not enabled on the analysis host, then essentially all files are limited to a maximum absolute pathname length of 259 characters. Directories may be limited to a maximum absolute pathname length of about 246 characters.

Win32 Long Paths Are Enabled

If win32 long paths are enabled on the analysis host, then many files are not subject to the maximum absolute pathname length limit of 259 characters. Here is a partial list of files that are still subject to this limit even if win32 Long Paths are enabled:

- Project Files.
- Project Archive Files.
- Ansys EDB (Electronics DataBase) files: These files are typically stored in the `ProjectName.aedb` folder.
- The temporary directory and most temporary files. Because some of the temporary files and temporary directories have automatically generated names that may be moderately long, it is **best to use a short pathname for the temporary directory**.
- Although most temporary files are created within the temporary directory, there may be some temporary files created within the project directory that are limited to a maximum absolute pathname length of 259 characters. As a result, the **project file directory pathname should be well below this limit**.

To enable Long Path support on Windows, both of the following requirements must be met:

- The application manifest must have the longPathAware setting specified as true, and
- Long paths must be enabled on the machine: There are two ways to do this; either may be used:

  Registry setting: `HKLM\SYSTEM\CurrentControlSet\Control\FileSystem : LongPathsEnabled=1`

  Via Group Policy Tool: The “Enable Win32 long paths” setting is in the folder: `Computer Configuration > Administrative Templates > System > Filesystem`

Related Topics

Saving a New Project
Saving the Active Project
Saving a Copy of a Project
Saving a Project as a Technology File

Saving a New Maxwell Project

1. Click **File>Save As**.
   The **Save As** dialog box appears.
2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** box.
   By default, all files have the **.aedt** extension.
4. Click **Save**.
   Ansys Electronics Desktop saves the project to the location you specified.

Warning	Be sure to save geometric models periodically. Saving frequently helps prevent the loss of your work if a problem occurs. Although Ansys Electronics Desktop has an "auto-save" feature, it may not automatically save frequently enough for your needs.

Related Topics

Saving the Active Project
Saving a Copy of a Project

Saving the Active Maxwell Project

- Click **File>Save**.
  Maxwell saves the project over the existing one.

Warning	Be sure to save geometric models periodically. Saving frequently helps prevent the loss of your work if a problem occurs. Although Maxwell has an "auto-save" feature,

	it may not automatically save frequently enough for your needs.

Related Topics

Saving a New Project
Saving a Copy of a Project

Saving a Copy of a Project

To save an existing, active project with a new name, a different file extension, or to a new location:

1. Click **File>Save As**.
2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** box.
4. Click **Save**.
   Maxwell saves the project with the new name or file extension to the location you specified.

Related Topics

Saving a New Project
Saving the Active Project

Save a Project as a Technology File

1. Click **File>Save As Technology File**.
   The **Save to user library** dialog box appears.
2. Click the **PersonalLib** button or **UserLib** button to specify the location to save the file.
3. Type the name of the file in the **File name** box.
   By default, all technology files have the **.asty** extension.
4. Click **Save**.

Related Topics

Saving a New Project
Saving the Active Project
Saving a Copy of a Project

Renaming a Project

To rename an existing, active project:

1. Select the project in the Project tree.
2. Right-click to display the short-cut menu.

3. Select **Rename**.

   This activates the text field for the project name.

4. Type the new project name and press enter.

   The new project name appears in the directory and the project remains in the original location.

Saving Project Data Automatically

Maxwell stores recent actions you performed on the active project in an Ansys Electronics Desktop auto-save file in case of a sudden workstation crash or other unexpected problem. The auto-save file is stored in the same directory as the project file and is named Project*n*.aedt .auto by default, where *n* is the order in which the project was added to the current session. After you have done a **Save As**, the autosave file name is <*name*>.aedt.auto, where <*name*> is the name you gave the project when saving.

Ansys Electronics Desktop automatically saves all data for the project to the auto-save file, except solution data. By default, Ansys Electronics Desktop automatically saves project data after every ten edits. An "edit" is any action you perform that changes data in the project or the design, including actions associated with project management, model creation, and solution analysis.

With auto-save activated, after a problem occurs, you can choose to re-open the original project file (Project*n*.aedt) in an effort to recover the solution data or to open the auto-save file. If a crash recovery file contains 0 bytes, the other file is opened instead.

To modify the auto-save settings:

1. Click **Tools>Options>General Options**.

   The **Options** dialog box appears.

2. Under the **General>Desktop Configuration** tab, verify that **Do Autosave** is selected.

   This option is selected by default.

3. In the **Autosave interval** box, enter the number of *edits* that you want to occur between automatic saves. By default, this option is set at 10.

Note	Auto-save *always* increments forward; therefore, even when you undo a command, Ansys Electronics Desktop counts it as an edit.

4. Click **OK** to apply the specified auto-save settings.

   Once the specified number of edits is carried out, a "model-only" save will occur. This means that Ansys Electronics Desktop does not save solutions data or clear any undo/redo history.

   When Ansys Electronics Desktop auto-saves, an ".auto" extension is appended to the original project file name. For example, "Project1.aedt " is automatically saved as "Project1.aedt .auto".

Warning	When you close or rename a project, Ansys Electronics Desktop deletes the auto-save file. Ansys Electronics Desktop assumes that you have saved any desired changes at this point.

Related Topics

Recovering Project Data in an Auto-Save File

Save Before Solving Option

The **Tools>Options>Maxwell 3D Options**, **Tools>Options>Maxwell 2D Options** or **Tools/Options/RMxprt Options** commands each display a dialog with a check box for an automatic **Save before solving** option. The main purpose is to force a full save before running the solve.

In the case where you start a solve while another solve is running, and the **Save before solving** option is set, Maxwell asks if you want solve without saving first. This lets you do multiple solves, and if you have not edited the project in between solves, crash recovery will work. In any case, you can start a new solve while running another without having to abort the running solve.

Recovering Project Data in an Auto-Save File

Following a sudden workstation crash or other unexpected problem, you can recover the project data in its auto-save file.

Warning	When you recover a project's auto-save file you *cannot* recover any solutions data; recovering an auto-save file means you will lose any solutions data that existed in the original project file.

To recover project data in an auto-save file:

1. If Ansys Electronics Desktop has crashed, launch Ansys Electronics Desktop from your desktop.
2. Click **File>Open**.
3. Select the original Project*n*.aedt project file for which you want to recover its Project*n*.aedt .auto auto-save file.

   The **Crash Recovery** dialog box appears, giving you the option to open the original project file or the auto-save file.
4. Select **Open project using autosave file to recover project data in the auto-save file, and then click OK.** Maxwell replaces the original project file with the data in the auto-save file.

   Ansys Electronics Desktop immediately overwrites the original project file data with the auto-save file data, removing the results directory (solutions data) from the original project file as it overwrites to the auto-save file.

Warning	If you choose to recover the auto-save file, you cannot recover the original project file that has been overwritten; recovering data in an auto-save file is *not* reversible.

Related Topics

Saving Project Data Automatically

Saving a Project From a Previous Version

When you open a Maxwell project from a version earlier than version 11, Maxwell creates a new project. The v10 or earlier project is not modified.

Archiving Projects

Use the **File>Archive...** command to “bundle” a project – and any other files related to the project that you want to include – in an **.mxwlz** file or **.zip** format archive. You can include notes about the contents of the archive and specify whether to include results and solution files, or related external files. The Archive command attempts to automatically detect the necessary files for linked projects and automatically include them in the archive. For example, if a project linked to the main project also has linked or associated files, you can add them.

Archive File Types

Internally, project archive files are .zip files, and are compatible with any program that can read .zip files (e.g. WinZip, 7Zip). The naming convention is that project archive files will have an extension that is unique for each product. The extension is generated by adding a 'z' to the project file extension (e.g..hfssz, .mxwlz). This extension will be displayed as the default when saving and restoring archive files. Note that .zip files are also included as a possible filter in file selection dialogs.

File Relocation Considerations

In a project to be archived, external files can be located anywhere on the user's system. One of the goals is for the restored project to be relatively self contained, and NOT to allow the restoration of an archived project to haphazardly write files anywhere on the restoring user's system.

To achieve this, it is sometimes necessary to change the location of files in the archived project such that the external files are now located in the project directory. At archive time, any external files not located in the project directory are relocated to the **restored_files** subdirectory of the project directory in the archived project. Any external files located in the user library or system library will be relocated to the personal library directory. Note that the project file that is written into the archive will be updated to refer to the files at the new locations, and the original project file will remain unaltered.

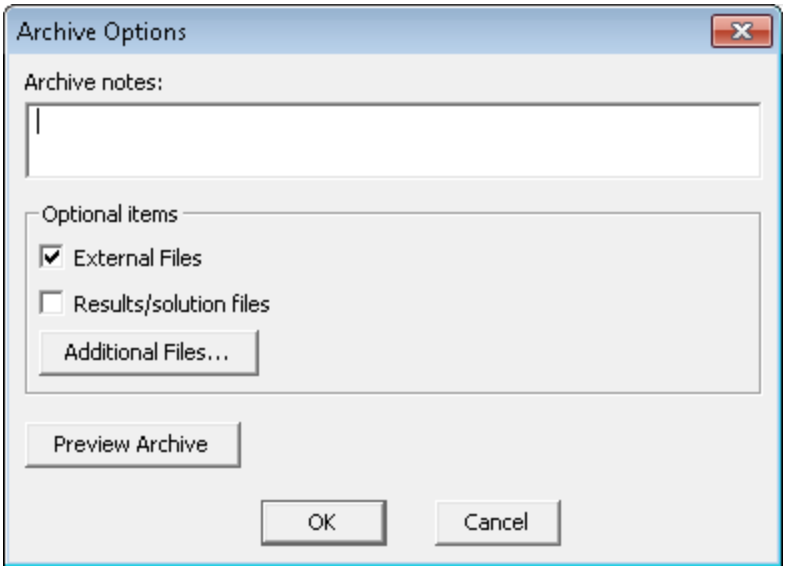
Archive Preview

The Archive command includes a preview feature that lets you review the contents of a planned archive.

To archive the current project:

1. Click **File>Archive...**

   The **Archive Options** dialog opens.

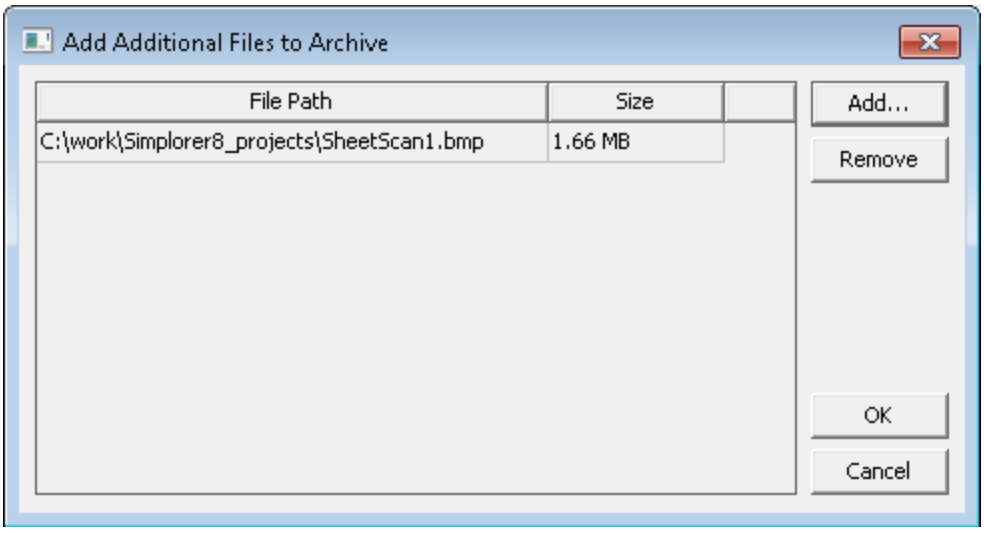


Archive notes: you can specify notes that will be visible when previewing the archive. These notes can be viewed from the preview dialog without actually restoring the archive.

External Files: selecting this check box will cause all external files to be included in the archive. The External Files check box refers to any existing files associated with the project, such as linked files, or files added through the **Project>Insert Documentation File** command or **Project>Data Set command**.

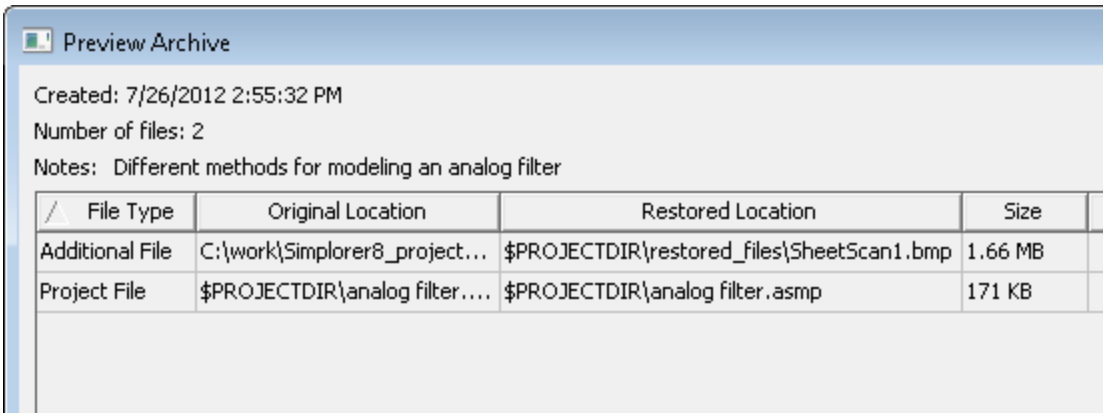
Results/solution files: selecting this check box will cause the entire results directory to be included in the archive. This may greatly increase the size of the archive file.

Clicking **Additional Files...** opens the **Add Additional Files to Archive** dialog.

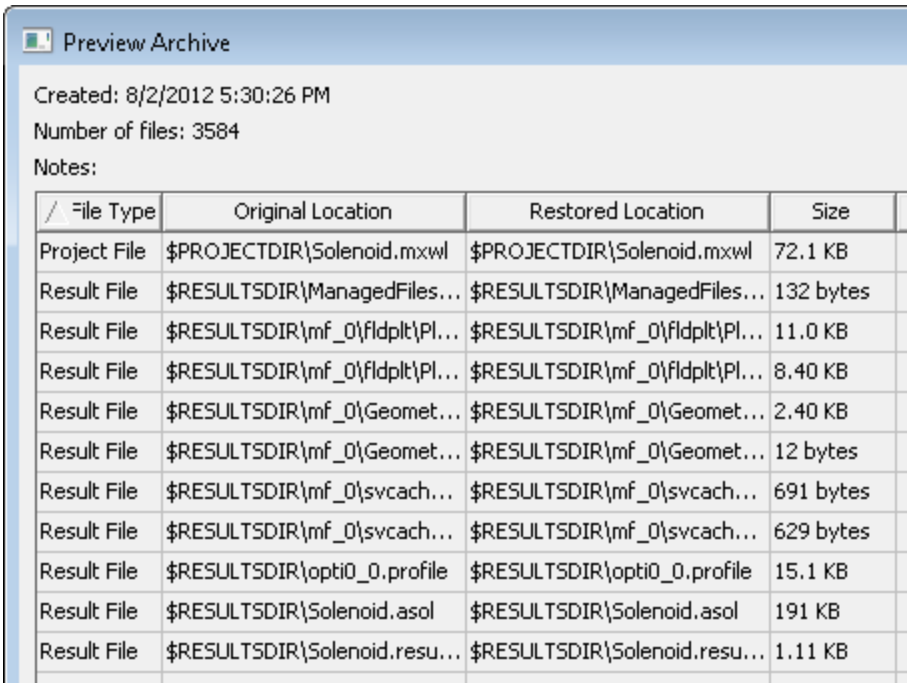


Here, you can click **Add...** to open a file browser to locate any additional files (such as documentation files or datasets) you want to include in the archive. You can also select and then **Remove** any files listed, and **OK** or **Cancel** any proposed changes.

2. Select any optional items, and make any desired archive notes in the text field.
3. When you have made your selections for optional items, you can select **Preview Archive** to look at the archive contexts, and the locations where restoring from the archive would place them.



Check the **Results/solution** option if you want to archive those files. Below is a preview showing some included results files.



To read longer locations, you can drag the column headers to expand the columns. Use the scroll bar to view longer lists.

Previewing an archive before creating the archive can be helpful in order to see exactly what files will be included in an archive, as well as how those files are being relocated. Another purpose of previewing an archive is to view warnings and consider if any additional files need to be added to the archive.

The preview dialog also displays the archive notes, creation date, and the number of included files.

4. When you are ready to create the archive, close the preview, and specify the format you want to use, **.mxwlz** or **.zip**, and specify the archive location and name. Click **OK** to create the archive.

Related Topics

Restoring Archives

Restoring Archives

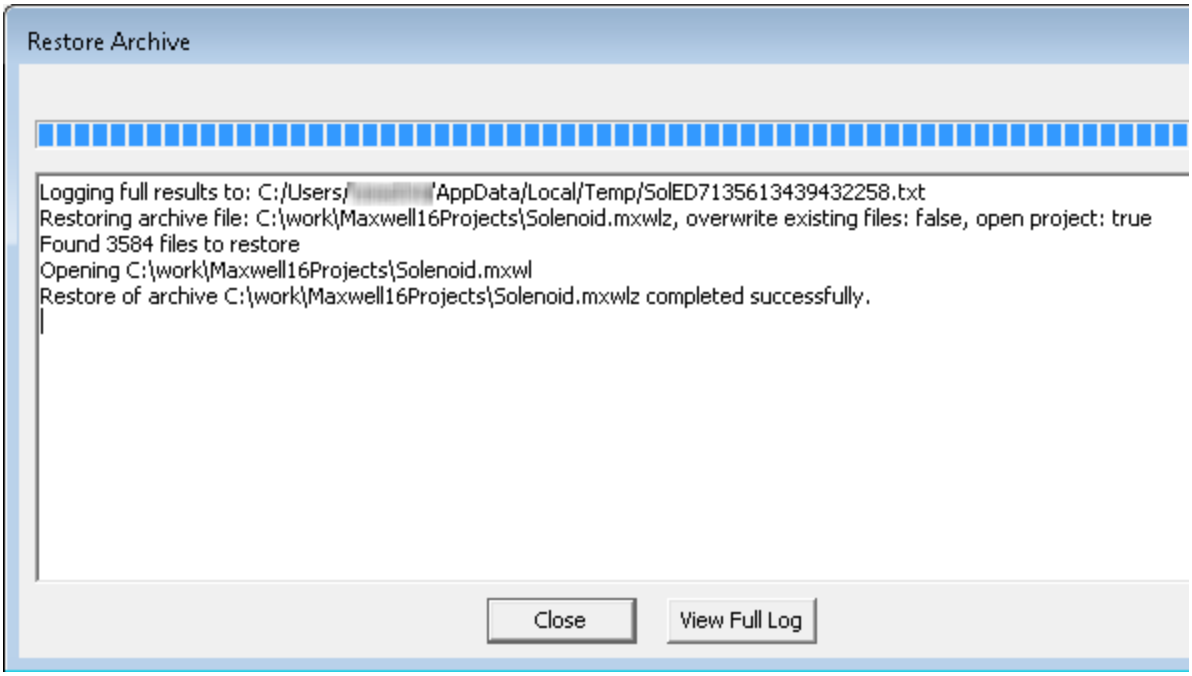
To restore an existing archive created with **File>Archive**:

1. Click **File>Restore Archive**

   This displays an **Archive to Restore** browser window that lets you navigate your file system for archive files of type **.mxwlz** or **.zip**.
2. After selecting a valid archive file, you can click **View Archive** to preview the contents.

   The preview dialog shows the same warnings that were generated at archive time. These warnings may be useful to identify additional steps that are needed to update any files to refer to files which had to be manually added to the archive.
3. Click **Open** to display a **Project File Restore Location** browser and navigate to the place where you want to restore the file.

   You can edit the file name, and check options to **Overwrite existing files** and to **Open project after restoring**.
4. Click **Save** to restore the archived file. A dialog displays progress and results of the restoration process.



A full log file is also generated which contains detailed information about the restore process. The first line in the text window displays the location of the full log file. After the restore has been completed, you can click **View Full Log** to display a detailed log of the restoration.

Related Topics

Archiving Projects

Closing a Project

To close a Maxwell or RMxprt project:

- Click **File>Close**.

If the project has not been saved since the last change, a message appears asking if you want to save the project before closing it. Do one of the following:

- Click **Yes** to save the project.
- Click **No** to close the project *WITHOUT* saving it.
- Click **Cancel** to keep the project open.

Deleting Projects

To delete a project:

1. Select the project in the project tree.
2. Click either **Edit>Delete**, or right click to display the short-cut menu and select **Delete**.

   A dialog displays the message: "The project selected and all its files will be deleted from the permanent storage medium. Click OK to proceed."
3. Click **OK** to delete the files or **Cancel** to retain them.

Exiting Maxwell and RMxprt

To exit the Maxwell and RMxprt interfaces:

- Click **File>Exit**.

Undoing Commands

Use the **Undo** command on the **Edit** menu to cancel, or undo, the last action you performed on the active project or design. This is useful for undoing unintended commands related to project management, model creation, and post-processing.

1. In the **Project Manager** window, do one of the following:
   - To undo the last action you performed on the *active project*, such as inserting a design or adding project variables, click the project icon.
   - To undo the last action you performed on the *active design*, such as drawing an object or deleting a field overlay plot, click the design icon.

Note	You cannot undo an analysis that you have performed on a model, that is, the **Maxwell>Analyze** command.
2. Click **Edit> Undo**, or click the **Undo** icon on the ribbons.

   Your last action is now undone.

Note	When you save a project, Maxwell always clears the entire undo/redo history for the project and its designs.

Related Topics

Redoing Commands

Redoing Commands

Use the **Redo** command on the **Edit** menu to reapply, or redo, the last action that was canceled, or undone. You can redo a canceled action related to project management, model creation, and post-processing.

1. In the **Project Manager** window, do one of the following:
   - To redo the last action you canceled on the *active project*, such as inserting a design or adding project variables, click the project icon.

- To redo the last action you canceled on the *active design*, such as drawing an object or deleting a field overlay plot, click the design icon.

2. Click **Edit>Redo**, or click the **Redo** icon on the ribbons.

   Your last canceled action is now reapplied.

Note	When you save a project, Maxwell always clears the entire undo/redo history for the project and its designs.

Related Topics

Undoing Commands

Removing Unused Components from a Project

To remove unused components from a project:

1. Click **Tools>Project Tools>Remove Unused Definitions**.

   The **Unused Definitions** dialog box appears.
2. Select the **Delete** check box for any components you want to remove.
   - You can also click the **Select All** button to select all components listed.
3. Click **Apply**.

   A warning message appears, telling you that the undo/redo history will be cleared for the project.
4. Click **OK** to continue, or click **Cancel** to cancel the deletion.
5. When you are finished removing components, click **OK** to close the **Unused Definitions** dialog box.

Updating Design Components

To update components defined in the current design:

1. Click **Tools>Project Tools>Update Definitions**.

   The **Update Definitions** dialog box appears.
2. Select one of the following two radio buttons.
   - Show Items with newer definitions
   - Show All Items
3. From the **Show Types** list in the **List Options** section, select the types of definitions you want to show in the **Item List** list.
4. Select the item you want to update from the **Item List** list.
5. Click the **Select All** or **Unselect All** buttons to select or clear all items listed.
6. Click **Update**.

   A message appears telling you the update was successful. Click **OK** to close the message.
7. When you are finished updating definitions, click **Close**.

Managing Projects and Designs

Copy and Paste a Project or Design

To **copy** a project or design:

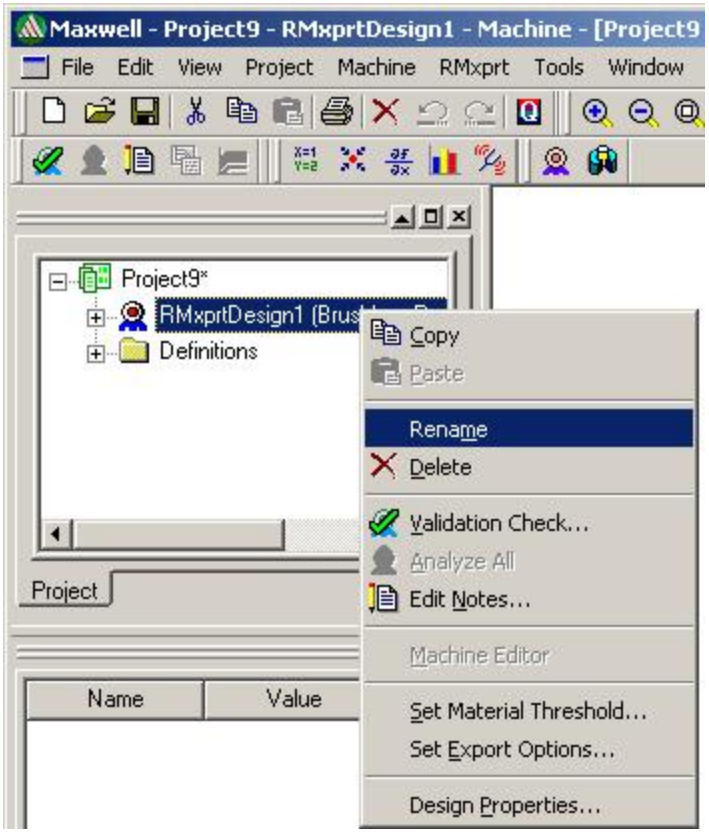
1. Select a project or design in the project tree in Project Manage Window to enable the menu command **Edit>Copy**.
2. Click **Edit>Copy**. The project or design is copied for pasting.

To **paste** a project or design:

1. Select a project or design in the project tree in Project Manage Window to enable the menu command **Edit>Paste**.
2. Click **Edit>Paste**. The project or design is pasted under the selected project, an icon is added to the project tree.

Rename a Project or Design

1. Right click the design icon to display the shortcut menu, as shown.



2. Click **Rename** in the shortcut menu.

3. Define the new name for the design by typing it directly into the Project Window.
4. Press **Enter** to complete the rename.

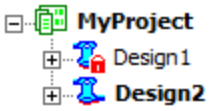
Delete a Project or Design

1. Select a project or design in the project tree in Project Manage Window to enable the menu command **Edit>Delete**.
2. Click **Edit>Delete**, or the **Delete** key.
3. Confirm the warning box to complete the delete operation, the icon is removed from the project tree.

Setting Read Only Designs

Designs can be set as either Read Only or Full Access. Full Access is the default status of a design and allows you to modify the design. In Read Only mode, you may only run the design or link it to another design. Setting a design as Read Only protects it from accidental modification.

Read Only designs are marked with a red lock in the Project Manager:



To change a single design's designation:

1. In the **Project Manager**, right-click the design.
2. Select either **Convert to Read Only** or **Convert to Full Access**.

To change the designation of *every design* in a project:

1. In the **Project Manager**, right-click the project.
2. Select either **Convert All Designs to Read Only** or **Convert All Designs to Full Access**.

> **Note:**
>
> Read Only settings are not saved when you save the project file.

Validating Designs

Before you run an analysis on a model, it is very important that you first perform a validation check on the design. When you perform a validation check on a design, Maxwell runs a check on all setup details of the active design to verify that the necessary steps have been completed and their parameters are reasonable.

To perform a validation check on the active design:

1. Click **Maxwell3D** or **Maxwell2D,** and then select **Validation Check**. You can also click the **Validate** button on the **Simulation** ribbon.

Maxwell checks the project setup, and then the **Validation Check** window appears.

2. View the results of the validation check in the **Validation Check** window.

   The following icons can appear next to an item:

	Indicates the step is complete.
	Indicates the step is incomplete.
	Indicates the step may require your attention.

3. View any messages in the **Message Manager** window.
4. If the validation check indicates that a step in your design is incomplete or incorrect, carefully review the setup details for that particular step and revise them as necessary.
5. Run a validation check after you have revised any setup details for an incomplete or incorrect design step**.**
6. Click **Close**.

Related Topics

"Validations Tab" on page 6-10

Exporting Files

You can export the following types of files from Maxwell:

- Ansoft 2D modeler files
- AutoCAD Drawing Interchange Format files
- 3D model files
- Graphics files
- Reports as data or graphics files in a range of formats

Related Topics

Exporting Matrix Data

Exporting Equivalent Circuit Data

Exporting 2D Modeler Files

When you export a 2D Modeler file, the geometry located within the xy plane is exported. You can choose to export only the selected parts of the geometry, or all of the parts.

Note	If you want to export a plane that does not coincide with the global xy plane, you must create a relative coordinate system to redefine the location of the origin and xy plane. See *Creating a Relative Coordinate System* for more information.

To export a file:

1. Click **Modeler**>**Export** .
   The **Export File** dialog box appears.
2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** box.
4. Select the desired model export format from the **Save as type** pull-down list.
5. Click **Save**. The file is exported to the specified location.

Related Topics

Exporting 2D AutoCAD Drawing Interchange Format Files
Exporting 3D Model Files
Exporting Graphics Files

Exporting 2D AutoCAD Drawing Interchange Format Files

When you export a file in the AutoCAD Drawing Interchange Format (**.dxf**), the 2D geometry located within the xy plane on the working coordinate system is exported. You can choose to export only the selected parts of the geometry, or all of the parts.

Note	If you want to export a plane that does not coincide with the global xy plane, you must create a relative coordinate system to redefine the location of the origin. See *Creating a Relative Coordinate System* for more information.

To export a file to a **.dxf** format:

1. Click **Modeler**>**Export**.
   The **Export File** dialog box appears.
2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** box.
4. Select **AutoCAD DXF Files(*.dxf)** from the **Save as type** pull-down list.
5. Click **Save**. The file is exported to the specified location as a **.dxf** file.

Related Topics

Exporting 2D Model Files

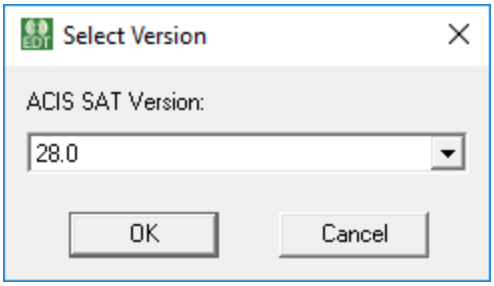
Exporting 3D Model Files

You can export 3D models to the 3D model file formats listed below. You can choose to export only the selected parts of the geometry, or all of the parts.

Type	Extension(s)
ACIS SAB	*.sab
ACIS SAT	*.sat
Ansys 3D Modeler	*.sm3
AutoCAD	*.dxf
CATIA	*.exp, *.model
GDSII	*.gds
IGES	*.iges, *.igs
OBJ Files (for importing models to Ensight.)	*.obj
Parasolid	*.x_t, *.x_b
STEP	*.step, *.stp

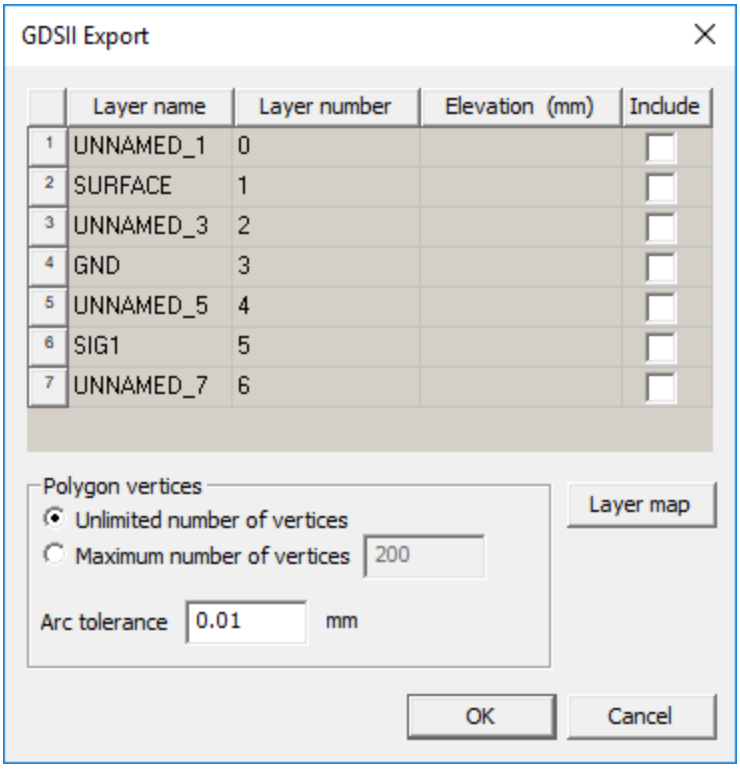
To export a file to a 3D model format:

1. Click **Modeler** > **Export**.
   The **Export File** window appears.
2. Browse to and select a location to save your file.
3. In the **File Name** field, name the file.
4. From the **Save as Type** drop-down menu, select the desired 3D filetype.
5. Click **Save**.
   The file is exported to the specified location with the appropriate file format.
   There is one additional option for *.sat files:



   - Use the **ACIS SM3 or SAT Version** drop-down menu to select an ACIS version.

   There are additional options for *.gds files:



- Use the **Include** check boxes to select layers for export.
- Click **Layer map** to define a *.layermap file, if desired. This is a text file that maps GDSII layer numbers to layer names in the stackup. The *.layermap file can have the same format as the .tech file used in GDSII import, but it only needs the layer name and number. Any other information is ignored.
- In the **Polygon Vertices** area, select either **No Limit to the number of vertices** or **Limit the number of vertices** and specify a value.
- For **Arc Tolerance**, leave the default or specify a value.

Related Topics

Exporting 2D Model Files

Exporting 2D AutoCAD Drawing Interchange Format Files

Exporting Graphics Files

Importing GDSII Format Files

Exporting Graphics Files

You can export the following graphics formats:

Extension	Contents
.bmp	Bitmap files.
.gif	Graphics Interchange Format files.
.jpeg	Joint Photographics Experts Group files.
.tiff	Tagged Image File Format files.
.wrl	Virtual Reality Modeling Language (VRML) files.

To export a file to a graphics format:

1. Click **Modeler**>**Export** to save the file in a graphics format.
   The **Export File** dialog box appears.
2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** box.
4. Select the desired graphics file format from the **Save as type** pull-down list.
5. Click **Save**. The file is exported to the specified location as a graphics file.

To export an image file to specified high resolution, use the scripting commands. Fonts and line thickness is not scaled. Only the image. You will have to iteratively increase font sizes till you find a suitable output.

```
oEditor.ExportModelImageToFile
("C:/Users/kmchrist/Documents/highresexample_image.jpg", 7680, 4320,
[
"NAME:SaveImageParams",
"ShowAxis:=" , "True",
"ShowGrid:=" , "True",
"ShowRuler:=" , "True",
"ShowRegion:=" , "Default",
"Selections:=" , ""
])
```

Related Topics

Exporting 2D Model Files

Exporting 3D Model Files

Exporting Data Table Files

You must have an existing plot open to see the **Report2D** menu.

1. Click **Report2D**>**Export to File**.

- Alternatively, right-click on the data table, and then click **Export to File** on the shortcut menu.

The **Export plot data to file** dialog box appears.

2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** box.
4. Select one of the following file formats from the **Save as type** pull-down list:

Extension	Contents
.txt	Post processor format file
.csv	Comma-delimited data file
.tab	Tab-separated file
.dat	Ansoft plot data file

5. Click **Save**. The file is exported to the specified location as a data table file.

Importing Files

You can import the following types of files to Maxwell:

- 2D model files
- 3D model files

Related Topics

Exporting Files

Importing 2D Model Files

You can import 2D model files directly into the active **Modeler** window:

Note	If you import a file into an active **Modeler** window that contains an existing model, the file is added to the existing model; it will not replace it.

If you want to import a model in a specific orientation other than the current XY or XZ plane, you must first create a relative coordinate system with the planes in the desired orientation. See *Creating a Relative Coordinate System* for more information.

To import a 2D model file:

1. Click **Modeler**>**Import**.

   The **Import File** dialog box appears.
2. Select a file type from the **Files of type** pull-down list.
3. Use the file browser to find and select the file you want to import.

4. Click **Open**.
   The file is imported into the active **Modeler** window.

Related Topics

Importing 3D Model Files

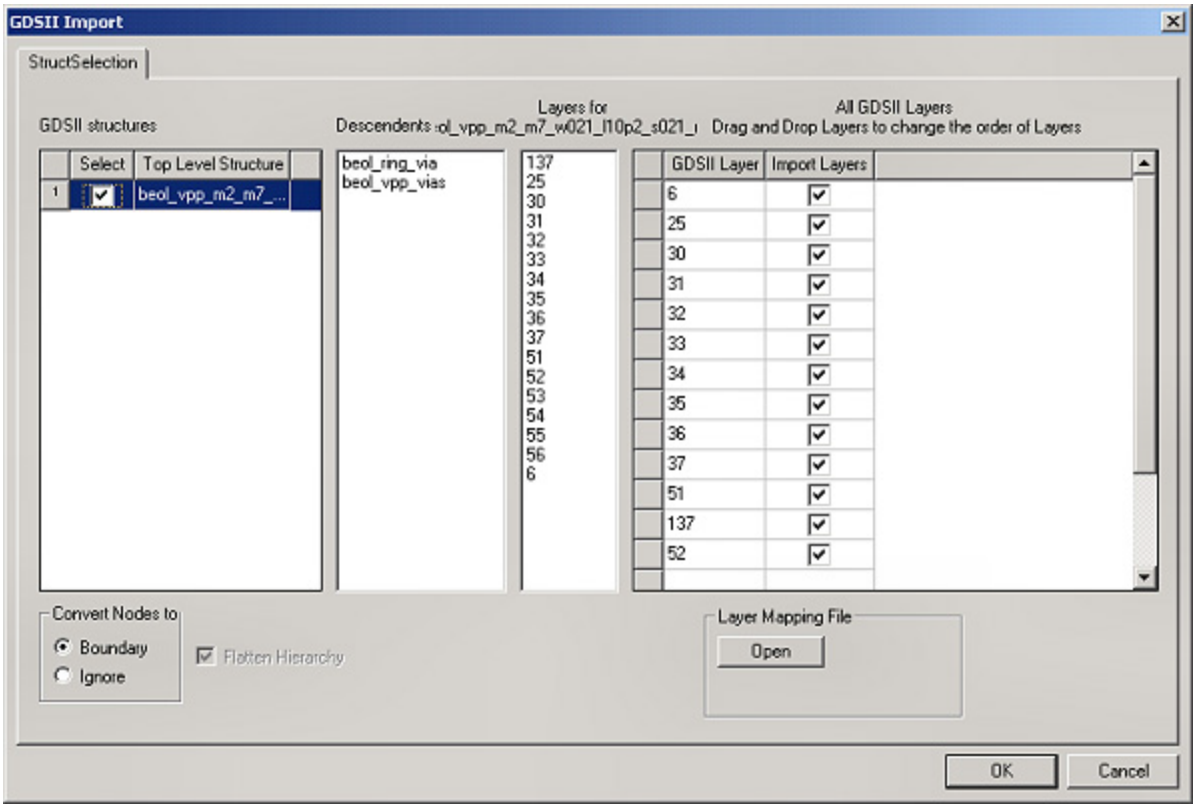
Importing GDSII Format Files

Importing GDSII Format Files

A GDSII format file is one type of 2D model file that can be read directly into the active Modeler window.

To import a GDSII format file:

1. Click **Modeler>Import**.
   The **Import File** dialog box appears.
2. Select **GDSII Files (*.gds)** from the **Files of type** pull-down list.
3. Use the file browser to find and select the GDSII format file you want to import.
4. Click **Open**.
   The **GDSII Import** dialog box appears.



5. Select the desired check boxes in the **GDSII structures** section of the dialog box.

   GDSII Structures Panel:

   The GDSII file may contain several top-level structures. You can do one of the following:

   - Click a structure name in the **GDSII Structures** panel to highlight it.
   - Clicking the **Select** check box in the **GDSII Structures** panel highlights the structure and selects that top-level structure to be imported.

   When multiple structures are imported, Maxwell creates multiple designs under the current project, one for each GDSII structure.

6. View the sub-layouts in the **Descendants** section of the dialog box.

   Descendants Panel:

   The GDSII file is hierarchical and may contain many sub-layouts. The **Descendants** panel shows the sub-layouts in the selected top-level designs.

7. In the **Layers for *structurename*** section, view the layers for the top-level structure.

   Layers for *structurename* Panel:

   The **Layers for *structurename*** panel shows the layers for the most recently highlighted top-level structure [*structurename*]. GDSII layers are identified by layer numbers.

8. In the **All GDSII Layers** section, select the layers you want to import.

   All GDSII Layers Panel:

   The **All GDSII Layers** panel lists all the layers from all the structures in the file.

   Use the **Import Layers** check boxes in the **All GDSII Layers** panel to select the layers to import. You can drag and drop the layers in the list to change the vertical stackup of layers.

9. In the **Convert Nodes to** section, convert or ignore objects that use the "nodes" data type.

   Convert Nodes to Panel:

   GDSII supports nodes and boundaries as separate data types. Normally, boundaries represent polygons. Maxwell can do one of the following: Convert objects that use the nodes data type to boundary types, or ignore them. You can do one of the following:

   - Use the **Convert Nodes to** radio buttons to select **Boundary** or **Ignore**. The default is to convert data type nodes to the data type boundary.
   - The **Flatten Hierarchy** check box is automatically selected. Maxwell always flattens any hierarchical geometry in the GDSII.

10. In the **Layer Mapping File** section, create a mapping to use for the import.

    Layer Mapping File Panel:

    If desired, you can create a mapping of the GDSII layer numbers to layer names in the design stackup. To create and use the mapping, do the following:

    - Use a text editor to create a text file that maps the GDSII layer numbers to layer names in the stackup. The layer mapping file must have a **.tech** suffix. The format of a **.tech** format layer mapping file lists includes the layer number and corresponding layer name, color, elevation, and thickness. For example, a layer mapping file may look like the following:

```
//#
//#  Layer        Color      Elevation        Thickness
//#  Name         Purpose       [nm]             [nm]
//#-------------------------------------------------------------------------
0    ref          red            0.000           0.000
17   POLYG        blue2        420.0           180.0
18   POLY2        blue2        420.0           180.0
25   PIMP         tan          400.0             0.000
26   NIMP         blue3        400.0             0.000
29   RPO          green        400.0             0.000
30   CONT         white        400.0           490.0
31   METAL1       red          890.0           280.0
```

- Click the **Open** button in the **Layer Mapping File** section to locate and open an existing layer mapping file.

11. When you are finished viewing and selecting options in the **GDSII Import** dialog box, click **OK**.

    The file is imported into the active **Modeler** window.

Related Topics

Importing 2D Model Files

Importing 3D Model Files

You can import 3D model files directly into the active **Modeler** window.

Supported 3D model file formats include:

Type	Extension(s)	Supported Version (s)
ACIS SAB	*.sab	ACIS R1 through ACIS 29.0.1
ACIS SAT	*.sat	ACIS R1 through ACIS 29.0.1
Ansys 3D Modeler	*.sm3	3D Modeler files in ACIS R2 through ACIS 29.0.1

Type	Extension(s)	Supported Version (s)
Ansoft Geometry	*.AnstGeom	All (see: UDMs and CAD integration with the Workbench.)
AutoCAD	*.dxf, *.dwg	Versions 2.5 (AC1002) through 2012 (AC1024) on Linux and versions 2.5 (AC1002) through 2017 (AC1027) on Windows (see: Importing DXF and DWG Format Files.
Autodesk Inventor	*.ipt, *.iam	V11 through V2022
CATIA	*.exp, *.model, *.CATPart, *.CATproduct	V4 through V5-6R2022
Creo Parametric	*.prt, *.asm	.Pro/E 16 through Wildfire 5.0, Creo 8.0
IGES	*.iges, *.igs	All, through 5.3
NASTRAN	*.nas	All, through 2018
Siemens NX	*.prt	NX 11 through NX 1980
Parasolid	*.x_t, *.x_b	9.0 through 34.0.153
SOLIDWORKS	*.SLDPRT, *.SLDASM	98 through 2022
STEP	*.step, *.stp	ISO standard 10303-21:2016, AP203 STEP files and AP214 (geometry only)
STL	*.stl	2.0

Note	
	• Object, material, and parameter names with non-ASCII characters are not allowed for data transfer. Such transfers fail and produce an error message. • If you import a file into an active **Modeler** window that contains an existing model, the file is added to the existing model; it will not replace it.

To import a 3D model file:

1. Click **Modeler>Import**.

   The **Import File** dialog box appears.
2. Select the file type you want from the **Files of type** drop-down menu.
3. Select any import options available for the selected file type. Option may include:
   - **Validation and Healing Options** – See Healing an Imported Object.
   - **Import Material Names** – use the check box to import material names.
   - **Import Free Surfaces** – for Creo Parametric files, use the check box to import surfaces as well as parts.
   - **Stitch Tolerance and Units** – For STEP and IEGS files, you can specify a Stitch tolerance and units. The default value (auto) comes from Healing options.
   - **STL Import Options** – for STL files for Modal or Terminal solutions, select the modeling units to which the imported model is scaled and whether to merge faces that are on the same plane, and whether to Heal on import. Heal attempts to remove common issues while importing stl files, including closing solids when they are open due to minor cracks in triangulation and removing self-intersections.

     The generic format specification for STL does not include units. When “Auto” is chosen for the file units, the current Model units are normally used. The exception is for *.stl files saved from SpaceClaim, for which units have been specified.

     Checking **Reduce** enables fields for specifying Reduce Max Error and Reduce Percentage.
   - **Create Groups for Sub Assembly** – for formats that allow importing MCAD Assemblies as groups, use the check box to retain the assembly structure of objects using groups. For every subassembly in the model, a group is created and it retains hierarchical information by creating group hierarchy. See: Group Commands for Modeler Objects.
4. Use the file browser to find the file you want to import.
5. Click **Open**.

   The file is imported into the active **Modeler** window.

For tips on dealing with very complex models, see Technical Notes: Handling Complicated Models.

Note	While objects created in Maxwell can always be classed in the history tree as either a solid, sheet, or wire some imported objects may have mixture of these. Maxwell places such objects in an **Unclassified** folder in the history tree.

Related Topics

Importing 2D Model Files

Importing DXF and DWG Format Files.

Exporting 3D Model Files

SpaceClaim Integration

Technical Notes: Handling Complicated Models

Importing DXF and DWG Format Files

You can import AutoCAD versions 2.5 (AC1002) through 2012 (AC1024) on Linux and versions 2.5 (AC1002) through 2017 (AC1027) on Windows. The entities are imported as 2D, not 3D. The types of entities imported are:

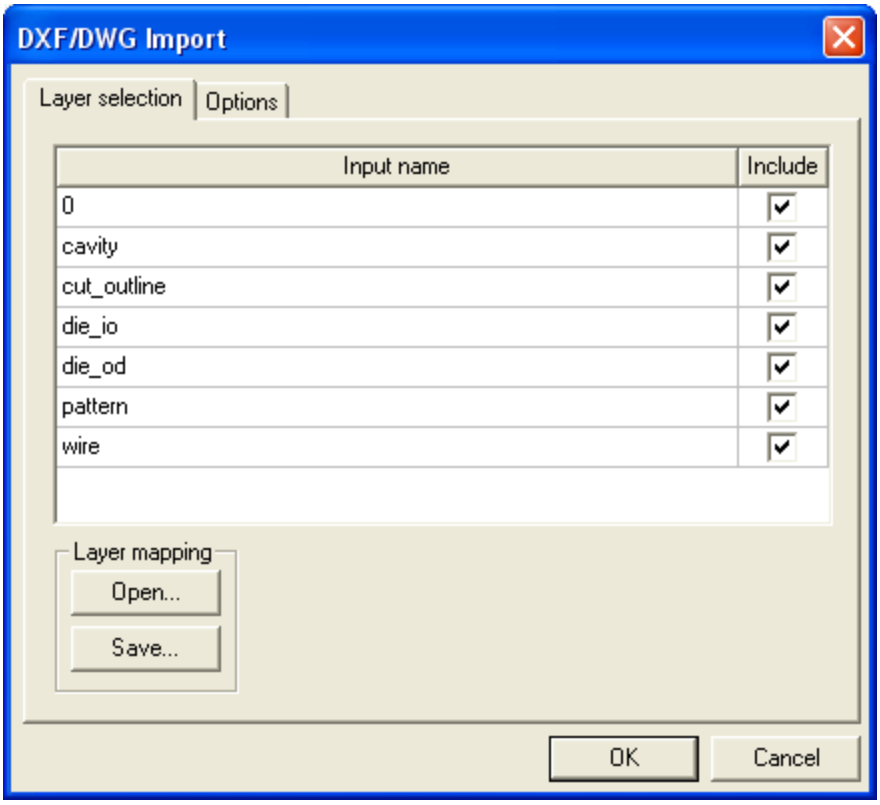
- 2D Polyline
- Polyline
- Line
- Arc
- Circle
- Ellipse
- Solid
- Block

To import a **.dxf** or **.dwg** model file (which may use an associated .tech file):

1. Click **Modeler**>**Import**.

   The **Import File** dialog box appears.
2. Select **AutoCAD Files (*.dxf;*.dwg)** from the **Files of type** pull-down list.
3. Use the file browser to find the file you want to import.
4. Select the **.dxf**/**.dwg** model file you want to import.
5. Click **Open**.

   Initially, the **DWG/DXF Import** dialog opens with the **Layer selection** tab displayed.

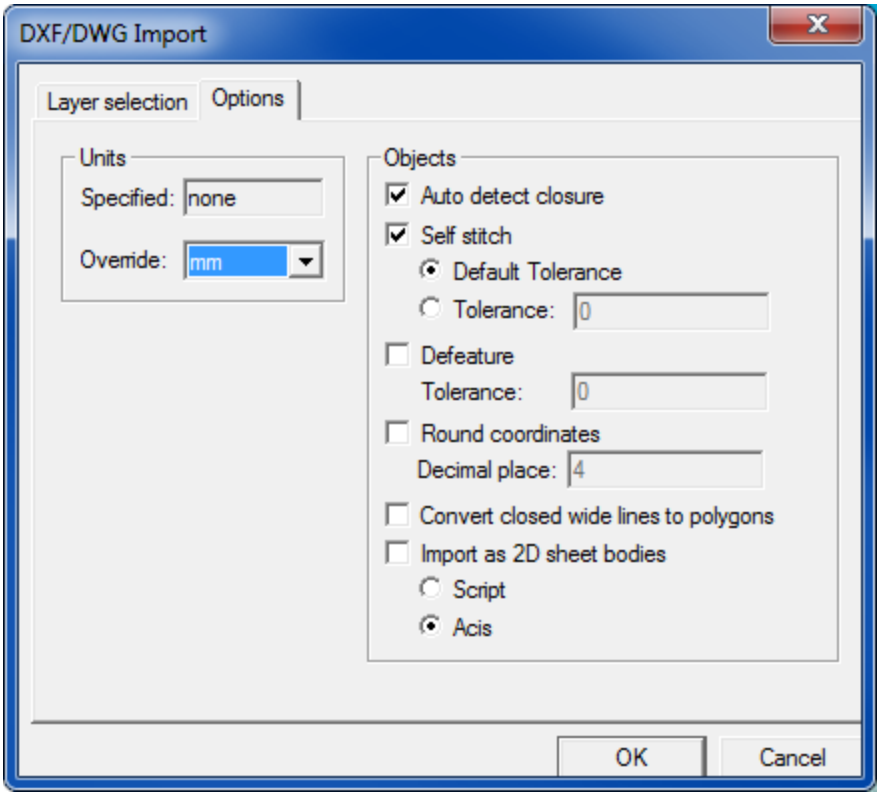


The **Input name** field shows the name of the layer in the DXF/DWG file (not editable).

6. Use the **Include** check boxes to specify which layers to import from the selected file.
7. You can use the browse button [...] to locate a Layer mapping file. The layer mapping file is a plain text file that includes units, layer names, color, elevation, and thickness information.

```
units um
//Layer_Name Color Elevation Thickness
BOTTOMLAYER purple 0 200
MIDLAYER green 500 200
TOPLAYER blue 1000 200
```

8. Click the **Options** tab.



9. Use the **Override** pulldown to select the layout units for the imported file (default is **mm**).
10. Use the Objects check boxes to fine-tune the import:
    - **Auto detect closure** causes polylines to be checked to see whether or not they are closed. If a polyline is closed, the modeler creates a polygon in the design.
    - **Self stitch** causes multiple straight line segments to be joined to form polylines. If the resulting polyline is closed, a polygon is created in the modeler.
        - **Default Tolerance/Tolerance** specifies whether you can specify a self stitch Tolerance value. If particular features in a model are outside of a normal tolerance allowance, you can set a Tolerance for importing that specific model.
    - **Defeature Tolerance** removes certain small features in the imported geometry to reduce complexity. The features that are removed include: multiple points placed within the specified distance; thin or narrow regions (“thins” and “spikes”); and extraneous points along straight line segments.
    - **Round coordinates** to **Decimal place** rounds all imported data to the specified number of decimal points.
    - **Convert closed wide lines to polygons** imports wide polylines as polygons. You have more flexibility to change the shape of such an object when it is imported as a polygon.
    - **Import as 2D sheet bodies** causes imported objects to be organized in terms of 2D sheets.

11. For Import method, select **Script** or **Acis**.
12. When you have completed selections on all tabs, click **OK** on any tab.

    The file is imported into the active **Layout** window.

Importing Data Tables

You can import data table files that contain data in the following formats:

- Tab-separated. Maxwell will recognize complex data if the values are separated by a comma (e.g. *real, imaginary*).
- Comma-separated. Maxwell will recognize complex data if the values are separated by a space (e.g. *real imaginary*).

1. Click **Maxwell**>**Results**>**Import Solutions**.
   - Alternatively, right-click **Results** in the project tree and then click **Import Solutions** on the shortcut menu.

   The **Imported Data** dialog box appears.
2. Click **Import Table**.

   The **Table Import** dialog box appears.
3. In the **File Name** text box, type the name of the data table file you want to import or click **Browse** and use the file browser to locate the file.
4. If the data in the table is complex, select the format — real/ imaginary, or magnitude/ phase — in which to import the data.

   If the data is simple, this option will be ignored.
5. Click **Load File**. Note that the file has not been imported yet.
6. Optionally, type a new name in the **Source Name** box that indicates the origin or the data table, or accept the default name.
7. Optionally, type a new name in the **Table Name** box that describes the data in the table, or accept the default name
8. In the **All Columns** list, the headings of each column in the data file are listed. Optionally, specify a new name for a column heading by doing the following:
   a. In the **All Columns** list, click the heading you want to change. The heading appears in the **Column Name** box.
   b. Type a new name in the **Column Name** box, and then click **Set Column Name**. The heading is changed to the new name in every place it appears in the **Imported Data** dialog box.
9. In the **Independent Data Columns** list, the first heading in the data table file is listed by default. In the **Dependent Data Columns** list, the second and subsequent headings in the data table file are listed by default. Optionally, click a heading name and then click an arrow button to move it from one column to another.
10. If the data in the **Dependent Data Columns** list contains matrix data, select **Matrix Data**. If it contains field data, select **Field Data**.
11. Click **Import**.

    You return to the **Imported Data** dialog box.

12. Click the data you want to import in the **Current Imports** list, and then click **OK**.
    The solution data is now available for post processing.

Related Topics

Adding Datasets

Inserting a Documentation File into a Project

To insert a documentation file so that it can be accessed from the project tree:

1. Click **Project>Insert Documentation File**.
   The **Open** dialog box appears.
2. Find and select the file you want to insert.
3. Click **Open**.
   The documentation file appears in the project tree for the current project.

Importing and Converting Materials

To preserve custom materials, you need to import them from a Maxwell **.mat** file (from a previous version of Maxwell) into the latest version of a Maxwell user library (**.amat** file).

To import these materials from previous versions, you must have at least one Maxwell design loaded in the project window.

To translate custom materials to the latest Maxwell version:

1. From the menu, click **Maxwell3D** or **Maxwell2D**, and then select **Translate Material Database**.
   The **Translate Legacy Material Database** appears.
2. Enter the name of the old database in the **Legacy Material DB Name** box. You can also click the **[...]** button to locate the database (**.mat** file).

Note	The **.mat** extension displays as a shortcut in the file browser.

3. Enter a name for the new database in your current project in the **User Material DB Name** box.
4. Click **Translate**. The dialog displays a Translation completed message when done. The imported materials are stored in a file with an **.amat** extension in the **userlib**.

Related Topics

Assigning Materials.

Importing a Model from the Clipboard

You can import a model from the Clipboard in order to use a geometry from a different design. To use a geometry with datalink, the geometry ID must be preserved. The **Import from Clipboard** command preserves geometry IDs.

To import a model from the Clipboard, the model for the current design must be empty. If the design it not empty, existing geometry is deleted.

The geometry model is imported from the Clipboard with the ID preserved.

1. Select the objects you want to copy. For selecting all objects, you can use **Edit>Select All** or **Ctrl-A**.
2. Click **Edit**>**Copy**  .

   The objects are copied to the Clipboard, a temporary storage area. The selected items are not deleted from the source location.
3. Select the design into which you want to paste the objects. It can be the same design from which you copied the items.
4. Click in the **3D Modeler** window.
5. Select the working coordinate system. Objects are imported relative to the current working coordinate system.
6. Click **Modeler**>**Import From Clipboard**.

The geometry model is pasted from the Clipboard with the ID preserved.

Related Topics

Setup Link Dialog

Selecting Items in the Modeler Window

Importing Tabular Arrays

Array data in tabular format can be imported from a file using the **Array Import** wizard that is invoked by selecting **Tools > Import Array from Table** and specifying a .csv file from the dialog that opens.

The tabular array .csv file should contain the following formatting:

```
arrIndex1 arrIndex2 arrIndex3
c1 "a0" d1
c2 "a2" d8
c3 "a3" d9
```

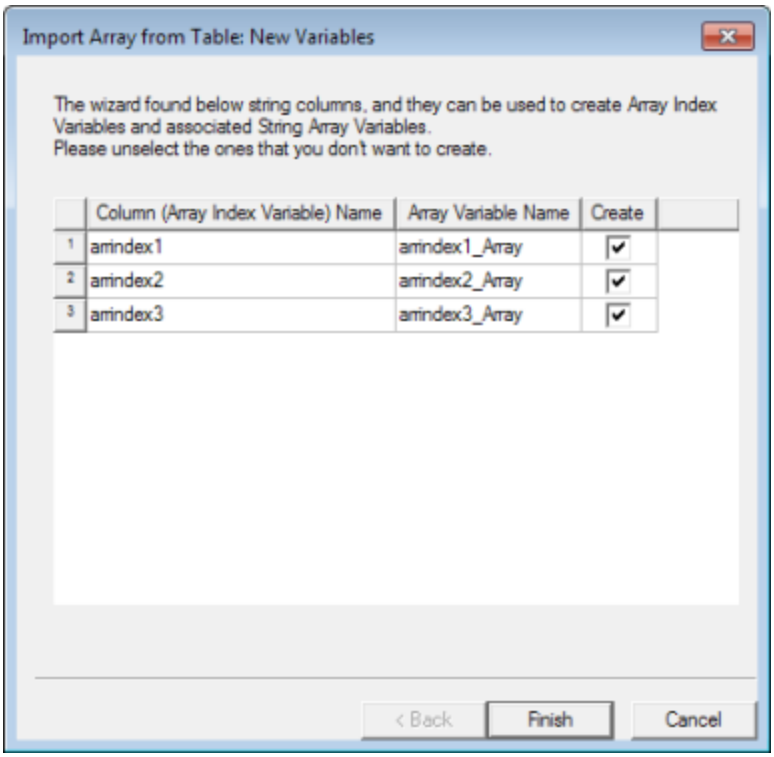
- First row consisting of list of Array Index Variable names separated by tabs or commas
- Remaining rows consisting of a list of string values separated by tabs or commas

- Double quotation marks (“ “) for string values are optional

After you select a .csv file to open:

- If no existing array variables are present in the design, the **Import Array From Table: New Variables** dialog opens.
- If one or more existing array variables are present in the design, the **Import Array From Table: Existing Variables** dialog opens.

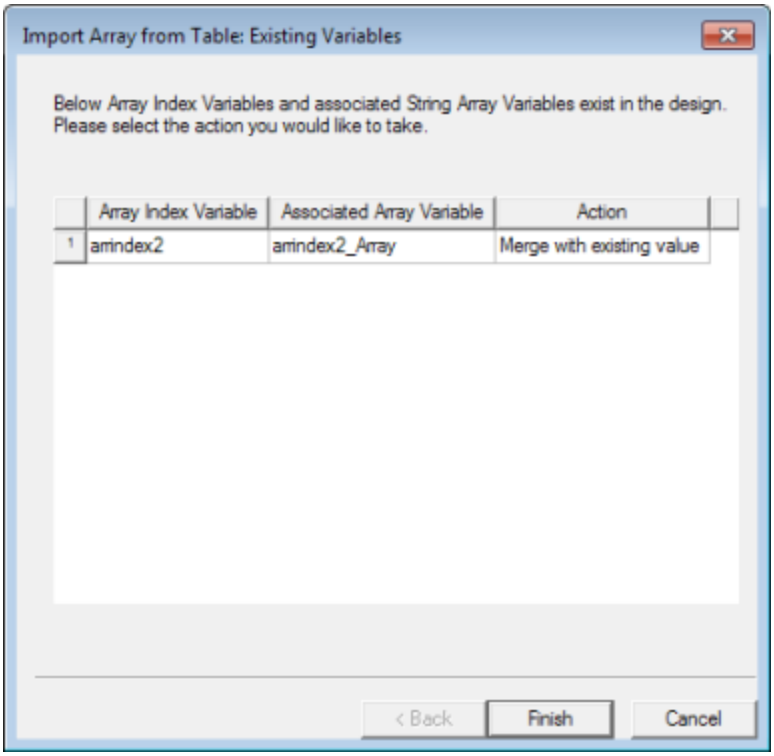
The **Import Array From Table: New Variables** dialog is shown below.



The following controls are available:

- **Column (Array Index Variable) Name** cannot be modified and is used to create the array index variable.
- The associated **Array Variable Name** has the default name “<*ColumnName*>_Array". You can modify the **Array Variable Name**, but variable uniqueness and validity is checked upon closing the wizard.
- Click the **Create** checkbox to specify that the array index variable should be created.
- Use the control buttons at bottom to go **Back** to the previous wizard dialog box, proceed to **Finish** and close the wizard dialog box, or click **Cancel** to close the dialog without making changes.

If one or more existing array variables are present in the design, when you click **Next** in the above wizard dialog box, the **Import Array From Table: Existing Variables** dialog opens.



The following controls are available:

- **Array Index Variable** name can *not* be modified.
- **Associated Array Variable** can be modified, but variable uniqueness and validity is checked upon closing the wizard.
- The following choices appear in the **Action** pulldown menu:

  – **Merge with existing value**

  – **Replace existing value**

  – **No Action**

  If you select **Replace existing value**, you are then prompted for verification.

- If you import a file to a design where a string column already exists as an **ArrayIndexVariable**, the value of the variable is checked when imported. If the existing array value is a superset of new array values, the array variable will not appear in the **ExistingVariables** dialog.

You can also use the same .csv tabular array file to setup a parametric analysis. When the parametric analysis is set up, the **ArrayIndexVariable** is searched and its index is used to create the table. Whether the **ArrayIndexVariable** refers to a double array or a string array, you must use the array element from the imported file - rather than the index value itself - to set up the **Parametric from File** analysis.

Printing

The printing commands enable you to send an image of the active window to the printer.

To print the project:

1. Click **File**>**Print**.

   The **Print** dialog box appears.
2. You can change the printer (if other printer names are listed on the drop down), set the print range, number of copies, or use the check box to Print to file.
3. Do one of the following:
   - Click **OK** to print the project.
   - Click **Cancel** to dismiss the window without printing.
   - Click **Properties** to define printer settings.

You can also access the printer properties by clicking **Printer** in the **Page Setup** dialog box.

Previewing the Printout

To preview how the page will look when printed:

1. Click **File>Print Preview**.
   The preview window appears.
2. To print the project after seeing the preview, click the **Print** button.
   The **Print** dialog box appears.
3. To navigate through the preview, click the **Next Page**, **Prev Page**, and/or **Two Page** buttons.
4. To zoom in or out on the preview, click the **Zoom In** or **Zoom Out** button.
5. To close the preview and return to your project, click **Close**.

Changing the Page Setup

To set or change the page setup:

1. Click **File>Page Setup**.
   The **Page Setup** dialog box appears.
2. Under **Paper**, select a **Size** and **Source** for the paper.
3. Under **Orientation**, select either **Portrait** or **Landscape**.
4. Under **Margins**, change the values as desired in the **Left**, **Right**, **Top**, and **Bottom** text boxes.
5. Click **OK**.

You can also access the printer properties by clicking **Printer** in the **Page Setup** dialog box.

Saving Design Notes

You can save notes about a design, such as its creation date and a description of the device being modeled. This is useful for keeping a running log on the project.

To add notes to a project:

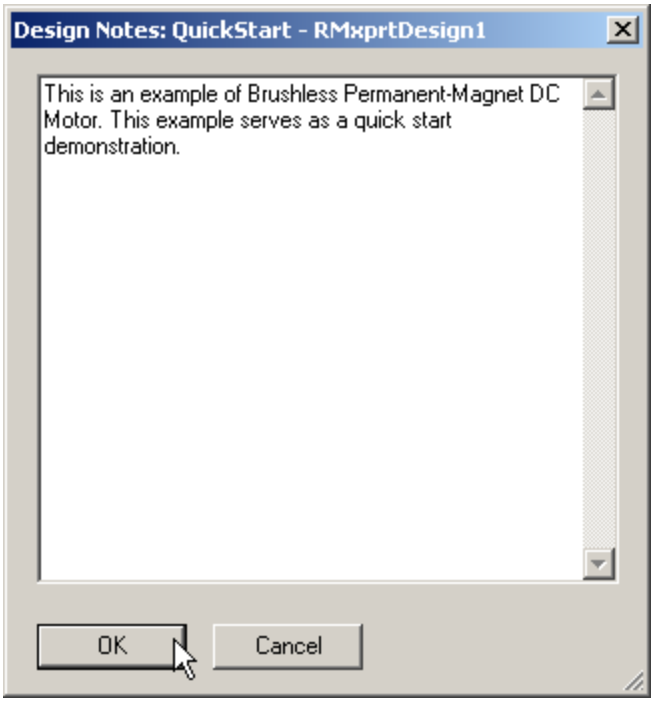
1. Click **Maxwell3D, Maxwell2D, or RMxprt, and then select Edit Notes**. Alternatively, right-click a design icon in the project tree and select **Edit Notes…** from the shortcut menu.

   The **Design Notes** dialog box appears.
2. Click in the window and type your notes.
3. Click **OK** to save the notes with the current project.

To edit existing design notes you may:

- Double-click the **Notes** icon in the project tree.
- Click **Maxwell3D, Maxwell2D, or RMxprt, and then select Edit Notes**.
- Right-click a design icon in the project tree and select **Edit Notes…** from the shortcut menu.

The **Design Notes** window appears, in which you can edit the design's notes as shown in the following example.



To delete the existing notes for a design:

1. Select the icon **Notes** in the project tree, click **Edit>Delete**.
2. Right-click the icon **Notes** in the project tree, click **Delete** from the shortcut menu.

The Notes icon is removed from the project tree.

Note	Notes are used to document aspects of designs only. For project level documentation, you can insert a documentation file into a project with the Project>Insert Documentation Files command.

Using the Password Manager to Control Access to Resources

Maxwell and RMxprt allow you to specify library resources that require password access and encryption. The same password can apply to multiple resources.

To access the Password Manager, click **Tools>Password Manager**.

Specifying a New Password Protected Resource

To specify a new password protected resource:

1. Click **Tools>Password Manager**.

   The **Password Manager** dialog box appears.
2. Click the **New** button.

   The **New Encrypted Resource** dialog box appears.
3. Specify the name of the resource that you want to protect, and click **OK**.

   The **Enter Passwords** dialog appears.
4. Select one of the following radio buttons:
   - **Enter password** - and confirm for **Full Access** or for **Execute Only Access**.
   - **Use Ansoft Password (Execute Only)** - This does not require you to enter a password, but it still encrypts the library.
5. After you have selected a radio button, and, if necessary, specified passwords correctly, click **OK** to return to the **Password Manager** dialog box in which the new resource is now listed.

Encrypting a Resource

To encrypt a resource:

1. Click **Tools>Password Manager**.

   The **Password Manager** dialog box appears.
2. Select an existing resource to highlight it and enable the **Encrypt File** button.
   A file browser window appears.
3. Select the desired **Circuit files (*.lib)** file you wish to encrypt.

   Any existing resources in the selected directory appear.
4. Click **OK**.

   The resource is encrypted.

Setting Options in Maxwell

You can use **Tools>Options>General Options** to open the **Options** dialog and set options for Maxwell as well as the Ansys Electronics Desktop. The following options are pertinent to Maxwell:

- General Options, such as project directory, WebUpdate options, and distributed analysis options.
- Maxwell 3D Options, such as the default solution type.
- Report Setup Options, including advanced mode editing, the number of significant digits to display, and drag and drop behavior.
- Maxwell 2D Options, such as the default solution type.
- RMxprt Options, such as the default solution type and material thresholds.
- Report2D options, such as formatting and cell size.
- Modeler options, such as coordinate system settings, color, and grid settings.
- Fields Reporter options, such as field overlay and phase animation settings.
- Schematic Editor options, such as fonts, colors ,etc. used by the Maxwell Circuit Editor.

You can use HPC and Analysis Options to specify design-specific options.

Additionally, the Tools>Options>Export Options Files command writes xml files containing the Options settings at all levels to the specified directory. This feature makes it easier for different users to use Ansys Electromagnetics tools installed on shared directories or network drives. The Example Uses for Export Options Features section outlines some use cases enabled by this feature.

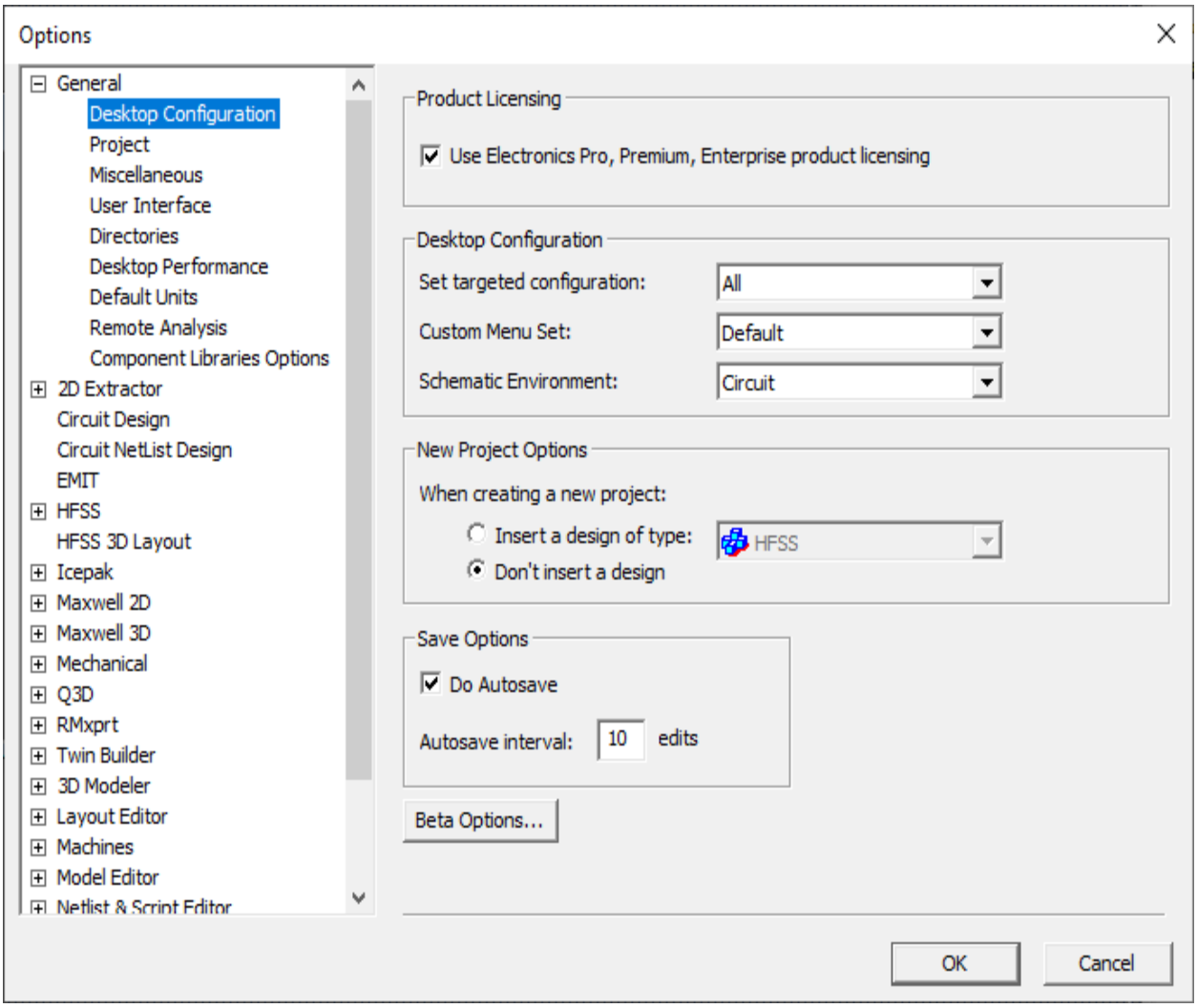
Setting General Options

To set general options in Maxwell:

1. Click **Tools>Options>General Options**.
   When the **Options** dialog appears, clicking the + next to **General** expands the list to show the following:
   - Desktop Configuration
   - Project
   - Miscellaneous
   - User Interface
   - Directories
   - Desktop Performance
   - Default Units
   - Remote Analysis
   - Component Libraries Options
2. Click each entry, and make the desired selections.
3. Click **OK** to apply your preferences.

General Options: Desktop Configuration

Under General Options, **Desktop Configuration** options allow you to customize Electronics Desktop in a way that suits your work priorities. These include options for product licensing, menus, new projects, save intervals, and beta features.



In the **Product Licensing** area, select whether to disable the Electronics Pro, Premium, Enterprise (PPE) product licensing mode. The PPE model is a portfolio of products that enable broad capabilities at each level, without requiring the purchase of add-ons and options. For example, features like Optimetrics, CAD translation, and circuit simulation are included directly in the PPE product levels.

In general, the Pro, Premium, Enterprise products use the same prep/post licensing as the traditional products. You can open and edit designs within the Electronics Desktop regardless of which product licensing has been selected. Simulation, and some other features like translation, use new licenses for the PPE products. If the **Use Electronics Pro, Premium, Enterprise**

product licensing option is not set appropriately these features will fail to check out their licenses.

When the PPE option is enabled, HPC licensing is used to enable all cores, GPUs, and distributed tasks. DSO is no longer used to distribute Optimetrics variations, as it was under the traditional product licensing model. ACL license sharing context manages license sharing of Electronics products. This allows design point (variation) distribution for external tools (WorkBench and optiSLang) to be based upon HPC licensing. A single set of solve and level licenses are checked out, instead of a set of solve and level licenses being checked out for each distributed design point. License sharing can be between applications (for example, Siwave and 3D layout), between designs (for example, datalink, cosim and dynamic link) and within a design (for example, distributed variations). The sharing context will always be used when the Pro, Premium, Enterprise (PPE) license model is active, and will never be used when PPE is not active. For more details, see Setting HPC and Analysis Options.

The **Desktop Configuration** section contains the following: **Set targeted configuration**, **Custom Menu Set**, and **Schematic Environment**.

- **Set targeted configuration** Each selection here also affects the default selections for the **Custom Menu Set** and **Schematic Environment** options (described below), as well as the default design type for a new project. Select the desired targeted configuration from the **Set targeted configuration** drop-down menu.
  - **All** – Default option. Sets the **Custom Menu Set** to **Default**, the **Schematic Environment** to **Circuit**, and the default design type to **HFSS**.
  - **EM** – Electromagnetic focus. Sets the **Custom Menu Set** to **EM**, the **Schematic Environment** to **Twin Builder**, and the default design type to **Maxwell 3D**.
  - **RF** – Radio Frequency focus. Sets the **Custom Menu Setto** RF, the **Schematic Environment** to **Circuit**, and the default design type to **HFSS**.
  - **SI** – Signal Integrity focus. Sets the **Custom Menu Set** to **SI**, the **Schematic Environment** to **Circuit**, and the default design type to **HFSS 3D Layout**.
  - **Twin Builder** – Sets the **Custom Menu Set** to **Twin Builder**, the **Schematic Environment** to **Twin Builder** and the default design type for a new project to **Twin Builder**.
- **Custom Menu Set** choices are:
  - **Default** – All solvers appear on the **Project** menu.
  - **EM** – Only electromagnetics solvers appear on the **Project** menu.
  - **RF** – Only radio frequency solvers appear on the **Project** menu.
  - **RF.0** – All solvers appear on the **Project** menu. The **HFSS RF Setup** menu appears.
  - **SI** – Only signal integrity solvers appear on the **Project** menu.
  - **SI1.0** – Only signal integrity solvers appear on the **Project** menu. The **Import**, **Solution Setup**, **Automation**, and **Definitions** menus appear.

- **SI2.0** – All solvers appear on the **Project** menu. The **HFSS SI Setup** menu appears.
- **Twin Builder** – Only Twin Builder appears on the **Project** menu.

NOTE: The default menu set selected depends on the selected targeted configuration.

- **Schematic Environment** – Select **Circuit**, **Twin Builder**, or **Maxwell**

New Project options specifies whether a design of a particular type is inserted when you create a new project. For example, if you define EM in **Set targeted configuration**, the default for **Insert a design of type** is Maxwell 3D. You can select other options from the drop-down list, or select the **Don't insert a design** option. The default is based on the selected targeted configuration.

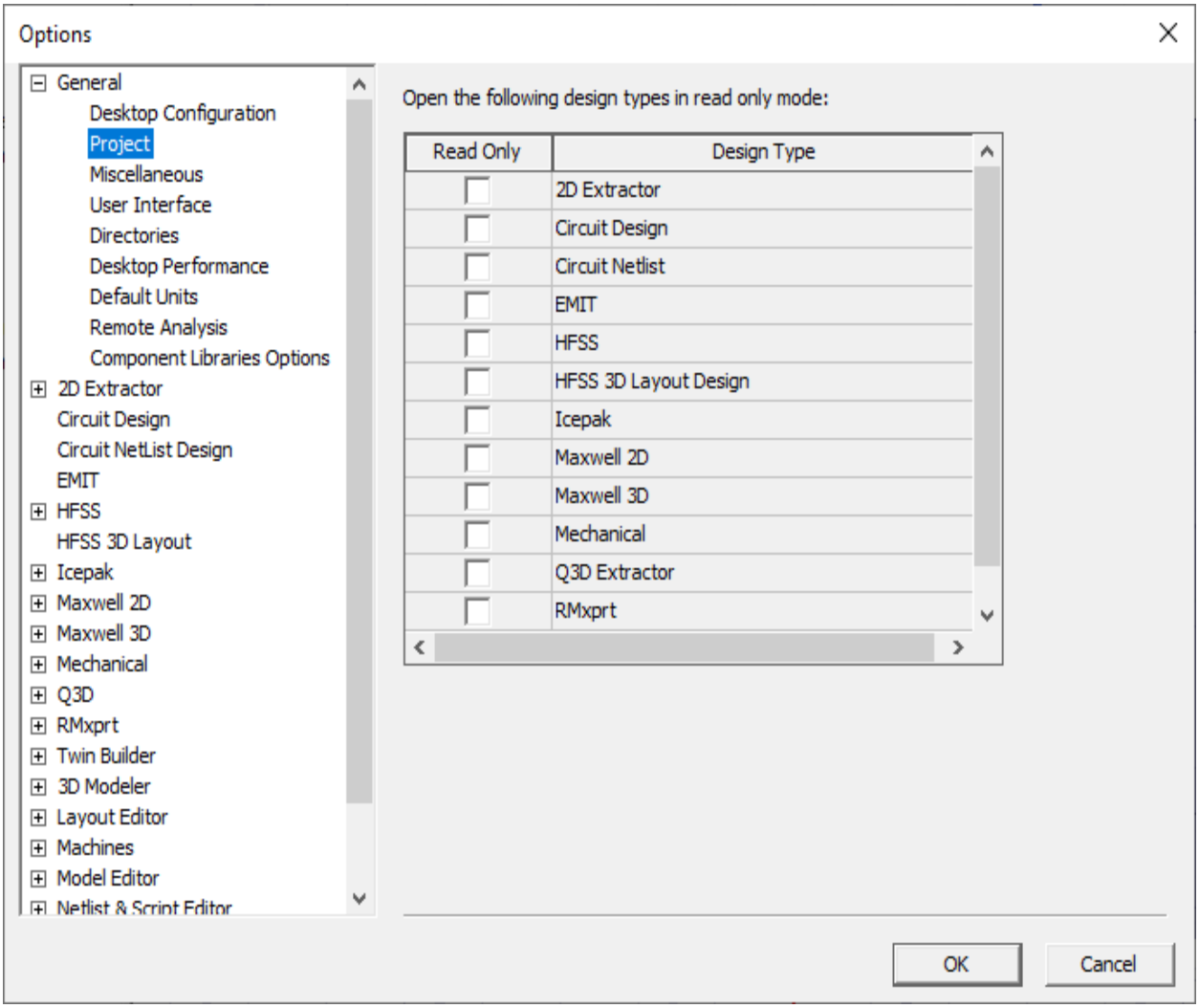
Save Options defines if autosave is enabled and the interval in edits for autosaving. To enable autosave, select the **Do Autosave** check box. Set the number of editing actions after which a backup file is automatically saved to your hard drive in the **Autosave interval** text box. The default is **10**. The backup file is given the same name as your project and has an "**aedt.auto**" file extension.

Beta Options opens a **Beta Options** dialog in which you can check any Beta Options you want to use, then click OK to close the dialog box. If you have selected any options, when you close the main Options dialog box, Ansys Electronics Desktop will prompt you to restart the desktop application to enable the selected beta features.

When all information on **Desktop Configuration** is defined, select another option under **General** in the **Options** dialog or click **OK** to put your settings into effect.

General Options: Project

These options are set on the **Project** panel under **General** in the **Options** dialog box. They allow you to select design types to open in read-only mode. This can help prevent accidental changes.



When all information on the **Project** panel is defined, select another option under **General** in the **Options** dialog or click **OK** to put your settings into effect.

General Options: Miscellaneous

These options are set on the **Miscellaneous** panel under **General** in the **Options** dialog.

Ansys Workbench Application

The Ansys Workbench Application **Path** lets you specify a path to an Ansys Workbench installation, if you have one. This path can be used to open the Icepak application when you create an Icepak design in RMxprt. It can also be used by the Optimetrics feature for connecting to the Design Xplorer.

MATLAB Optimization

If you have an installation of MATLAB installed you can use it as an Optimizer. This MATLAB path setting must to point to the version of MATLAB to be used for performing the optimization. .

Note	The platform (64 bit) of the specified version of MATLAB must match the platform of this application.

Save Preview Images

Click **Save preview images in project file** if you want the images to be saved in your projects.

When all information on **Miscellaneous** is defined, select another option under **General** in the **Options** dialog or click **OK** to put your settings into effect.

General Options: User Interface

These options are set on the **User Interface** panel under **General** in the **Options** dialog.The **User Interface** panel contains the following sections:

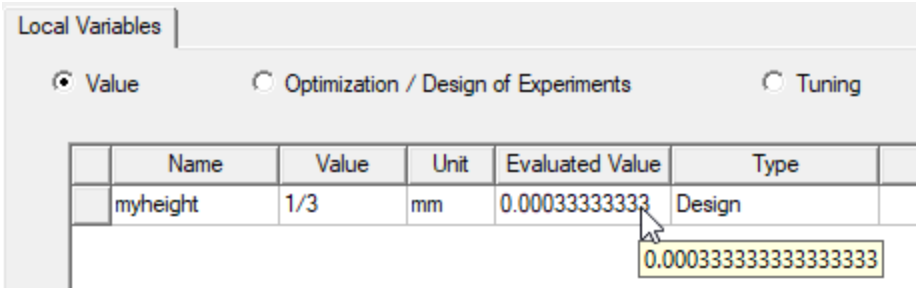
General

- **Show Message Window on new messages** – when selected, the message window automatically opens if a message arrives.
- **Ensure that new messages are visible in the Message Window Tree** – when selected, the message window expands as needed to display messages.
- **Show Progress Window when displaying progress bars** – when selected, the progress window automatically opens while simulations are in progress.

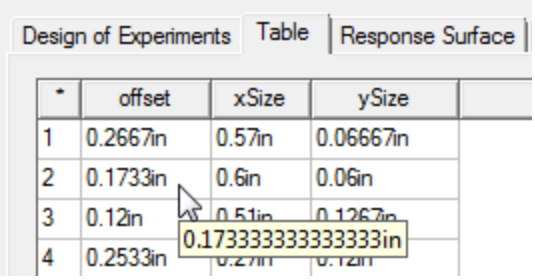
Project Tree Visualization Options

- **Emphasize active command context** – when selected, active elements in the Project Tree display in bold text.
- **Change icon when selection does not match active window** – when selected, a small, window-shaped overlay icon displays in the corner of the selected Project Tree element. This icon changes when the data in the active window is unrelated to the selected project item (data affecting the same model is considered to be related). Clicking the icon opens the window and brings it into focus.
- **Expand Project Tree on Insert** – when selected, the Project Tree automatically expands when you insert a new design.

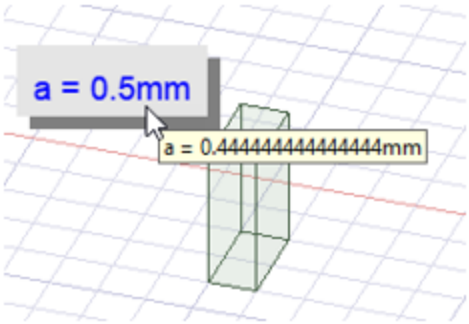
In the **Display Options** area, specify the **Maximum Number of Significant Digits** to display. The default is 8 and the maximum is 20. This affects the digits displayed in the Solutions dialog box, evaluated variable values, Animation dialogs, Optimetrics, Reports, and so forth. You can still see the full precision values in the tooltips by holding a cursor over the displayed value.

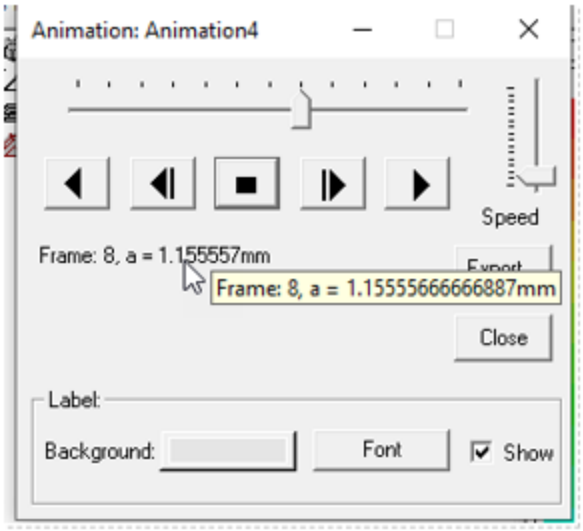


In the case of variable values, if you have assigned more significant digits, you will see these when editing the variable value. In the case of table displays of values, the tooltip display shows all available digits when the mouse pointer is over a result:



The tooltip functions to show internal digits in throughout the Ansys Electronics Desktop interface, including the Model window, animation dialogs and so on.





Color Scheme

You can select the color scheme of Blue, Silver, or Black.

Welcome Message

Select **Show welcome message at startup** if you want to see a welcome message when Electronics Desktop opens.

When all information on the **User Interface** panel is defined, select another option under **General** in the **Options** dialog or click **OK** to put your settings into effect.

General Options: Directories

1. For each directory, Project, Syslib, UserLib, and PersonalLib, enter a path in the **Directory** text box, or click the ellipsis button ("**...**") to find and select the desired directory. If you modify SysLib, UserLib or PersonalLib paths through **Tools>General** options, the User Defined Primitives menu should reflect those changes on next startup or on **Draw>User Defined Primitive>Update Menu**.
2. If you want to enter a directory path in the **Temp Directory** text box, you must check Override to enable the path text field and the ellipsis button ("**...**") for finding and selecting the desired directory.
3. To reset the library directories to the default, click **Reset Library Directory**.

General Options: Desktop Performance Tab

These options are set on the **Desktop Performance** panel under **General** in the **Options** dialog box.

Report Update Options for Design Type:

- Select a **Design type** to which the report update options apply.
- Select or clear **Dynamically update reports and field overlays during edits**.

  If selected, report plots and overlays update dynamically.
- **Dynamically update postprocessing data for new solutions**.

  Updating numerous reports may take a significant amount of time. Updating reports during the analysis process can impact the overall time to solution. You may want to vary the times when your reports get updated relative to the impact on overall solve time.
- **Update reports on file open –** specifies that reports be automatically updated whenever an existing file with solution data is opened for viewing/editing. Enabled by default.

  Five options exist for updating reports during solutions:
  - **Automatically** - the default. It means update most things immediately.

    For “AdaptivePass” plot context, plots are updated at the end of each solution pass. For “LastAdaptive” or “Transient” the plot is updated at the end of the transient or adaptive solution. This option balances report and field plot updating with solution time. For example, reports may be updated after each adaptive pass but field plots will not be updated until the solution is complete.
  - **Immediately** - update reports and plots as soon as data comes from the solver.

    This option will have the greatest impact on the overall solution time but will have the most rapid updating of reports and field plots. Caution should be used in selecting this option. Some types of reports and field plots may take a long time to update, especially as the mesh size increases.
  - **Never** - only manual intervention updates reports.

This option will prevent updates from impacting the solution time.

- **On Completion** - as with **Never**, but a single update is done when the solve completes.

Note	Reports that are updated on completion are done after the solve has been completed. The time for that update is not included in the solve profile.

- **After Each Variation** - when performing an Optimetric or parametric analysis, all reports are updated after analysis of each variation has been completed.

Animation

Computing animated plots requires significant memory which depends upon the complexity of the plot type. The animation setting is used to prevent problems related to low memory should an animation require large memory allocation.

The Animation setting lets you set a limit to **Stop computing animation frames when available memory is less than** the set value in megabytes. The default is 100.

Desktop Pre/Post Processing settings:

- **Number of Processors** – The default is determined by the number of logical processors on the machine. This option is common for both pre-processing and post-processing. This option only affects pre/post processing in desktop (not solve or simulation).

  More details regarding the default core usage.

  - The default core usage per desktop session (UI + solve) is set to 2/3 of the logical processors on the machine.
  - The cores for default local config is max{4, 1/3 of the logical processors on the machine}
  - The default number of processors for pre/post is min{1, 2/3 of the logical processors - default_cores_for_local_config}
  - When the core usage per desktop session cannot be evenly distributed between solve and pre/post, more cores will be assigned to solve.

  Example 1:

  - logical processors = 24
  - cores usage per desktop session = 16
  - local HPC set to 4 cores
  - processors for pre/post set to 12

  Example 2:

  - logical processors = 20
  - cores usage per desktop session = 13
  - local HPC set to 4 cores
  - processors for pre/post set to 9

  Example 3:

  - logical processors = 5
  - cores usage per desktop session = 3
  - local HPC set to 2
  - processors for pre/post set to 1

The following pre-processing algorithms can take advantage of multiple processors:

- Visualization/faceting of 3D models for 3D products
- Model validation for 3D products
- Auto net identification for Q3D only

Disk Space Warning

You can set a value to **Warn when available disk space is less than** the specified number of gigabytes.

When all information on the **Desktop Performance** panel is defined, select another option under **General** in the **Options** dialog or click **OK** to put your settings into effect.

General Options: Default Units

These options are set on the **Default Units** panel under **General** in the **Options** dialog.

The **Default Units** panel of the **General Options** dialog box allows you to set the default unit values for the following metrics: **Length**, **Frequency**, **Resistance**, **Angle**, **Power**, **Inductance**, **Time**, **Voltage**, **Capacitance**, **Temperature**, **Current**, **Force**, **Torque**, **Speed**, **Angular Speed**, **Magnetic Induction**, **Weight**, **Mag**netic **Field Strength**, **Pressure,** and **Conductance**.

Note	Units specified in the **Default Units** panel are generally for problem definition. The Post Processor displays output in the unit that scales best for plotting, however, the user may change the unit when creating the plot. Also, the Fields Calculator always works with SI units.

When all information on the **Default Units** panel is defined, select another option under **General** in the **Options** dialog or click **OK** to put your settings into effect.

General Options: Remote Analysis

Under General Options, **Remote Analysis** options allow you to launch all analyses as a service or specified user rather than as the current user.

In the **RSM Service Options** area, options include:

- **Ansoft Service Port** – Click **Change** to update the port number. Ansys Electromagnetics RSM Service should be running on this port for all distributed machines.
- **Send Analysis Request As** – Select either **Service User** or **Specified User**. Selecting **Specified User** enables the **User Name**, **Password**, and **Domain/Workgroup** fields.

  > **Note:**
  >
  > If any of the remote machines is Unix-based, you must specify the current user.

- **Disable Access By Remote Machines** – If desired, select to disable access for remote machines.

When multiple IP addresses are available, the **Desktop-Engine Connection** area allows you to specify the preferred IP address for communication:

- **Use Default** – your system's default IP address.
- **Specified Address** – an IP address you specify.
- **Specified Subnet** – a subnet you specify. Subnet may be network prefix and prefix length (123.12.123.0/22), network prefix and subnet mask (123.12.123.0/255.255.252.0), or network prefix only (123.23.123.0).
- **Use Loopback Address for Local Solves** – modifies how Electronics Desktop connects to the simulation engine when solving on a local machine. Selecting this option can make the connection more reliable and avoid interruptions associated with connecting and disconnecting with VPN.

Changing the Listening Port used by Ansys RSM Service

To change the listening port used by the RSM Service, you must change the ansoftrsmservice.cfg file as follows:

Specify the ListenPort within a 'CommDetails' block, which must be within a 'Default:CommDetails' block, which must be within the top level block of the file (the 'AnsoftCOMDaemon' block). The following example changes the listen port from 32958 to 32957, with these blocks at the beginning of the file:

```
    $begin 'AnsoftCOMDaemon'
$begin 'Default:CommDetails'
  $begin 'CommDetails'
  ListenPort='32957'
  $end 'CommDetails'
$end 'Default:CommDetails'
. . . .
    $end 'AnsoftCOMDaemon'
```

For the second level block, ensure that there is a single colon character and no spaces or tabs separating the two parts of the block name 'Default:CommDetails'. The third level block, with name 'CommDetails' is also required. Use caution when editing this file by hand, because any typos in the block or value names may cause the data to be ignored.

Related Topics

Remote Analysis

General Options: Component Libraries Options

These options are set on the **Component Libraries Options** panel under **General** in the **Options** dialog.The **Component Libraries Options** panel contains the following sections:

You can select to **Use extractor mode by default for separate dynamic link desktops** by using the check box.

You can select to **Run affected property callbacks when variable changes value** by using the check box.

The **Component Tree Options** panel contains the following controls:

- **Show Favorites** specifies that a list of favorites, in the form of an expandable tree, is displayed in the Component Libraries's Components tab window.
- **Show Most Recently Used** specifies that a list of the components you have most-recently used is displayed in the project window.
- **Most Recently Usedlist contains** specifies the number of recently used components to display.

The **Search Options** panel contains the following controls:

- Use **Set** to determine whether searches are performed on **All components**, **Current list only**, or **Append to current list**.
- **Load Libraries at Initialization** – When checked, Twin Builder pre-loads all system and configured libraries into the Component Libraries search database when the Component Libraries is initialized at Twin Builder startup. Clear this option to reduce the startup time of Twin Builder, in which case the libraries are loaded at first search.

When all information on the **Component Libraries Options** panel is defined, select another option under **General** in the **Options** dialog or click **OK** to put your settings into effect.

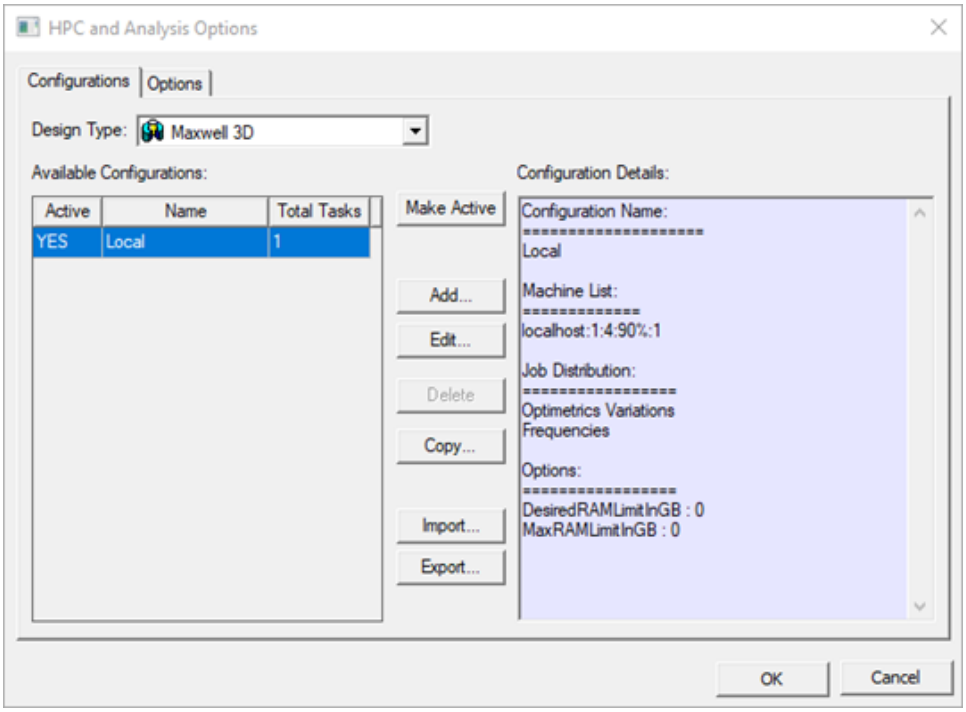
Setting HPC and Analysis Options for Maxwell and RMxprt Designs

HPC and analysis parameters are accessed via a single dialog. Machine list and options settings are integrated into analysis configurations. The default configuration is for solving on a single, local machine. You can create many analysis configurations for remote and distributed solutions, and switch between them, depending on the job being solved. Multiprocessing is integrated into the machine lists.

To set the HPC and Analysis Options:

1. Click **Tools>Options>HPC and Analysis Options.**

The **HPC and Analysis Options** dialog appears, displaying two tabs.



Configurations tab

In the **Configurations** tab, you can select the Design Type, and select from a list of available configurations. Selecting a Design Type displays a list of the Available configurations for that type. Selecting a configuration from the list displays the details of that configuration in the text fields. A Name can describe the use for which a configuration has been defined. The Total Tasks column shows the number of tasks that the analysis configuration can execute.

Design Type

Defines configurations for the selected design type. If you want to use similar analysis parameters for a different design type, you must create a separate analysis configuration for that design type. The Active configuration is used when solving an analysis for the design type.

Available Configurations List

From the lists of Available Configurations for each design type, click the desired configuration in the list to select it, then click **Make Active**. The active configuration is indicated with a YES in the Active column.

- **Add...** button – launches a dialog to create a new analysis configuration.
- **Edit...** button – launches a dialog to edit the currently selected analysis configuration.
- **Delete** button – deletes the currently selected analysis configurations.

Note	You cannot delete the Local configuration.

- **Copy...** button – creates a new analysis configuration, and launches a dialog to edit it. If the dialog is canceled, the new analysis configuration is not created.
- **Import...** button – allows the user to import an .acf file to create an analysis configuration.

Note	Importing analysis configurations always adds the imported analysis configurations to the current design type. Also, if there is a name conflict between an imported analysis configuration and an existing analysis configuration, the imported configuration is renamed and you are notified.

- **Export...** button – allows the user to export the selected analysis configurations to an **.acf** file. Users can then import the configurations into a different design type, or import them on a different machine.

Options Tab

Note	Options for Maxwell 2D, Maxwell 3D, and RMxprt design types are described below. Options for other design types, such as HFSS, are described in the *Ansys Electronics Desktop Help*.

The Options tab in the **HPC and Analysis Options** dialog contains design type specific options. These options are not part of an analysis configuration: instead they are always in effect for the given design type when the following is true:

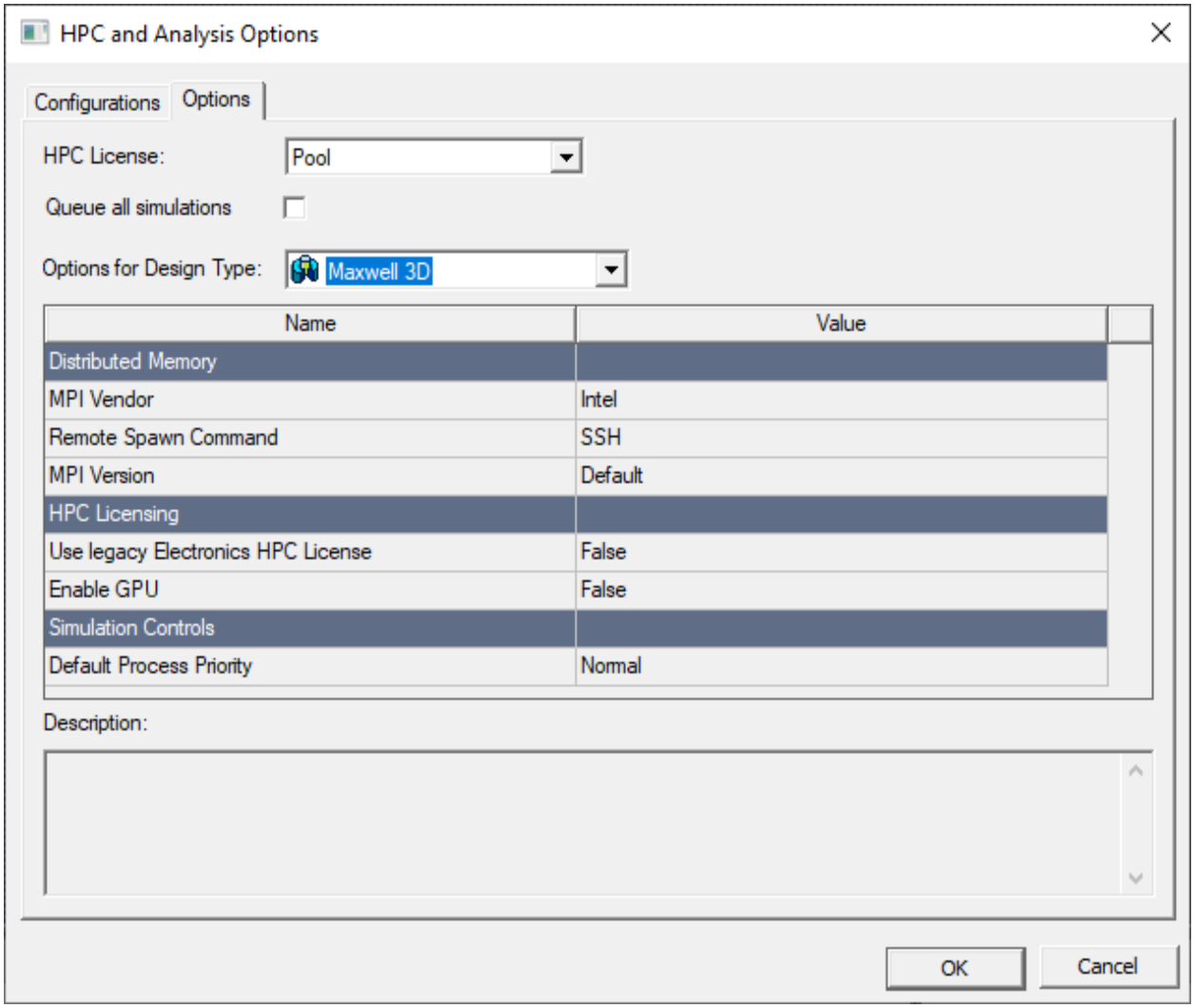
- A design of the matching design type is being solved
- You have not specified corresponding overriding batch options on the command line.

In the **Options** tab you can enable queuing by clicking the **Queue all simulations** check box, and set the design type.

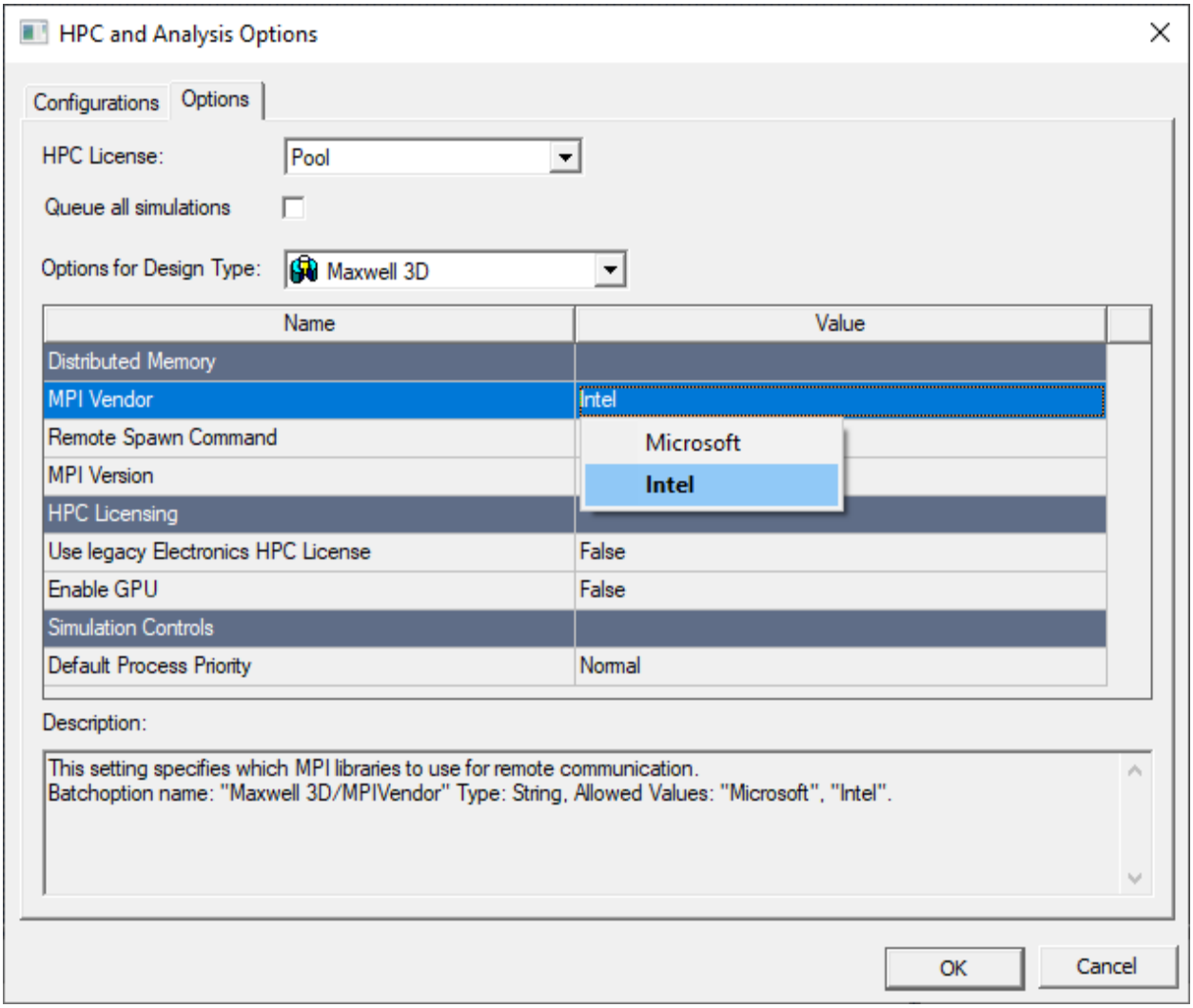
- For Maxwell 2D and 3D design types, you can specify the Distributed Memory MPI (Message Passing Interface) vendor, specify the Remote Spawn Command, set the HPC license type, and set the Default Process Priority.
- For the Maxwell 3D design type only, you can also Enable GPU (if installed) for Eddy Current solves.
- For the RMxprt design type, you can only set the Default Process Priority.

Solving on a single Windows machine does not require MPI installation; and users running on Linux do not need to install MPI manually.

If the **Queue all simulations** check box is selected, the Desktop queues any active simulations for design types that have **Save before solving** turned off in **General Options** and then processes them in order. You can view and change the queue by using Show Queued Simulations.



2. For Distributed Memory options, use the drop-down menu to select the MPI Vendor for the selected Design type.



The solvers use the standard Message Passing Interface (MPI), and can perform solutions that distribute memory use across machines in a cluster or network. Memory used by the MPI-enabled solver is therefore limited by the set of machines that are available rather than the shared memory available on any single machine. This allows you to simulate larger structures, and to optimally reconfigure the cluster of machines for the problem at hand. For solving on a single machine, MPI is not required, nor does it provide an advantage.

To use the distributed memory solution you must install MPI software from one of the supported third party vendors on all the machines you intend to use.

You may need to set passwords depending on the MPI vendor for authentication on the machines. Settings within each design type turn on distributed memory solutions and define the list of machines you intend to use.

You can also specify the MPI version. If not specified, then the default version for the MPI vendor will be used. This setting is ignored if there is only one supported version for the selected MPI Vendor. Multiple versions are only supported for Intel MPI, as follows:

- The value "Default" indicates that the default Intel MPI version should be used. This is Intel MPI 2018 in most cases.
- The value "2018" indicates that Intel MPI 2018 should be used.
- The value "2021" indicates that Intel MPI 2021 should be used.

InfiniBand Support for Windows

By default, the MPI vendors use the fastest interconnect (typically InfiniBand is faster than Ethernet). If you want to override the default behavior and force the use of Ethernet, you can set the **ANSOFT_MPI_INTERCONNECT** environment variable to **eth** for the job.

3. For Linux authentication, you can specify the Remote Spawn Command as RSH or SSH (the default).
4. For **HPC License** select either the **Pool**, **Pack** or **None** license option.

   HPC licensing enables the use of cores and GPUs to accelerate simulations. In general, each core requires one unit of HPC, while each GPU requires eight units. The selected HPC license type determines which license is used, and how units of HPC are converted to license counts.

   - **Pool** - One HPC pool license enables one unit of HPC.
   - **Pack** - One HPC pack license enables eight units of HPC. Additional packs multiply by four, enabling 32, 128, 512, etc., in the context of a single simulation.
   - **None** – Disables the use of HPC in some scenarios (see below). Since the Electronics Pro, Premium, Enterprise (PPE) products (the default in General Options: Desktop Configuration) use HPC to enable all parallel and distributed simulations, the **None** option is not shown under **HPC License** when using PPE.

   The Electronics Desktop products include four units of HPC for each licensed simulation. This means that up to four units can be used without requiring HPC licenses; license counting will begin with the fifth unit. For example, a simulation that uses 36 cores requires 32 HPC units after subtracting the four included cores. This simulation will check out 32 HPC pool licenses, or two HPC pack licenses.

   When using traditional Electronics product licensing, variations are distributed via the Distributed Solve Option (DSO). Distributing N variations requires N-1 licenses of DSO. The HPC units are then calculated on a per variation basis. In other words, the HPC licenses will enable HPC for a single variation, and then DSO extends this HPC to all distributed variations. If you do not have HPC licenses, choose **None** under **HPC License**. You can still use DSO to distribute variations, with up to the four included HPC units per variation.

   A significant advantage of the Electronics Pro, Premium, Enterprise products is that the HPC products enable all parallel and distributed simulations, *including distributed variation*s. As a result, DSO licenses are not compatible with the PPE products.

   When the PPE option is enabled, HPC licensing is used to enable all cores, GPUs, and distributed tasks. DSO is no longer used to distribute Optimetrics variations, as it was under the traditional product licensing model. ACL license sharing context manages license sharing of Electronics products. This allows design point (variation) distribution for external tools (WorkBench and optiSLang) to be based upon HPC licensing. A single set of solve

and level licenses are checked out, instead of a set of solve and level licenses being checked out for each distributed design point. License sharing can be between applications (for example, Siwave and 3D layout), between designs (for example, datalink, cosim and dynamic link) and within a design (for example, distributed variations). The sharing context will always be used when the Pro, Premium, Enterprise (PPE) license model is active, and will never be used when PPE is not active.

In prior releases, a distributed parametric with a HPC config of [3 tasks, 16 cores] always required 20 hpc-pool, even when there are only 2 variations to be solved. With ACL context, license checkouts are on-demand and distributing 2 variations with [3 tasks, 16 cores] only checkouts 11 hpc-pool. With ACL, enabled by PPE, required licenses are held for the full course of a simulation. The value of this improved behavior can be amplified in a cluster simulation of large Optimetrics.

For externally driven solves: the driving application creates a context and all instances of Electronics Desktop run under that context. This means that PPE and solve licenses will only incur a single checkout. HPC checkouts are distributed and done on demand.

For WB - Each instance is counted as a single variation using a single core. Behavior should be the same as in earlier versions.

For LSDSO - A context is passed through different levels of desktopjob down to the ansysedt.exe. Behavior should be the same as earlier versions with regard to license counts, except HPC checkouts are distributed and done on demand instead of being done in a single checkout.

For OptiSlang - Single PPE and solve license checkouts, and HPC checkouts distributed and done on demand.

For Remote solves: remote solves on different machines can all run under the same context. PPE and solve licenses incur a single checkout and HPC checkouts are done on demand.

For Twin Builder - PPE is not supported by Twin Builder. However, you can still select to use PPE licensing in an Electronics design (for example, Maxwell or HFSS) that is coupled to a Twin Builder design. When coupled with a Twin Builder simulation, electronics designs are expected to share its PPE licenses. For example, during a distributed variations simulation of the Twin Builder design, the electronics solve features that are requested in parallel should ey supported by HPC Parametric (8 ans_hpc per variation) instead of one copy of the solve feature per variation. For appropriate sharing of ans_hpc (pool and pack), Twin Builder stops using DSO to distribute variations when “use PPE” is selected.

For HPC Pool, distributing *N* variations requires 8**N* pool licenses and enables up to four HPC units per variation. Each additional *N* pool licenses will enable one additional HPC unit per variation. For HPC Pack, distributing *N* variations requires *N* pack licenses and enables up to four HPC units per variation. Each additional *N* pack licenses will enable 8, 32, 128, etc., additional HPC units per variation. Since the PPE products use HPC to enable all parallel and distributed simulations, the **None** option is not shown under **HPC License** when using PPE.

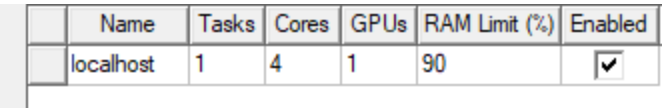
Solution setups with domains, such as transient designs using the Time Decomposition Method, always use HPC licensing. Even though domains are distributed and may use multiprocessing, they do not use distributed solve or multiprocessing licenses – all of this is included in the HPC licensing when solving domains.

When variations are not distributed, core capping is done by including the first four cores from the machine/task entries in the user specified hpc setup. If the first entry has four or more cores, then only that machine/task is included, with its cores capped at four. If the first entry has fewer than four cores, all of its cores are included, and cores from additional entries add in the same manner until the overall cap of four cores is reached.

Note	Do not select **None** if the simulation requires more than four cores HPC. For **None**, see step 5.

5. If you choose to opt out of HPC licensing by selecting None, the Ansys Electronics Desktop enables four cores for all solvers. The four cores are not limited to a single node, meaning that features such as distributed frequencies, DDM, and TDM can all be used without HPC licenses, as long as the total core count is four or less. The four cores scale with DSO. For example, a user with 2 DSO can distribute 2 variations with 4 cores each, for a total of 8 cores, without requiring HPC licenses.
6. The **Use legacy Electronics HPC License** option defaults to **True**, meaning that legacy HPC licenses are checked out by default. These licenses are reported in the log. If the option is set to **False**, then the anshpc and anshpce_pack licenses are checked out.
7. For 2D projects, select the **License model for distributed variations** as either **DSO** or **Solver**. The Optimetrics variations can be distributed by the solver licenses. One solver license will be consumed for each variation being solved in parallel.
8. Optionally, users with one HPC pack or 8 HPC pool licenses, and who have an NVIDIA® graphics card installed, can use GPU acceleration for Maxwell 3D Eddy Current designs. For details on the requirements for GPU use, see Using GPU Acceleration for Maxwell 3D Eddy Current Designs.

   To enable use of GPU acceleration, set **Enable GPU** to **True** in the HPC and Analysis Options dialog; and set **Use Automatic Settings** to **off** in the Analysis Configuration dialog box. The matrix solver automatically determines if all cores should be used, or if one GPU should be used to give the best performance. For example, if you specify 4 cores for the simulation, the 3D Eddy current solver will use 4 cores in parallel during matrix assembly while the matrix solver will use either 4 cores or 1 GPU.



	Name	Tasks	Cores	GPUs	RAM Limit (%)	Enabled
	localhost	1	4	1	90	☑

You can determine if a GPU is being used for acceleration by viewing the Solutions dialog box, Profile tab. If a GPU is successfully locked for use by the solver process, the profile shows the GPU's CUDA device ID and its name.

If the GPU is not used, the Profile indicates that, and the fallback to using the CPU cores.

Enable/Disable GPU Acceleration from the Command Line

GPU acceleration can be toggled by the -batchoptions command line argument:

```
EnableGPU =[0/1]
```

For example, the following command turns on GPU acceleration:

```
ansysedt.exe -batchsolve -batchoptions "Maxwell 3D/EnableGPU' =1"
projectname.aedt
```

When the GPU is Used:

- GPU must be enabled for Maxwell.
- Both Windows and Linux are supported.
- Only complex symmetric matrices can be solved by CUDA.
- Matrix must be large. By default, its dimension should be larger than 2,000,000.
- The times using CUDA and CPU only are estimated. If the GPU is faster, it will be used. If not, the solver will fall back to multi-core CPU.

Time Estimation for GPU and CPU Usage:

- Time estimation is based on the structure of the matrix, not just the dimension.
- The generation and model of GPU are considered. The newer the model is, the faster the GPU estimation is.
- The clock rate and number of cores of CPU are considered. The higher the rate is, and the more the cores there are, the faster the CPU estimation is.
- The faster device (either GPU or CPU) will be selected based on the estimated time.

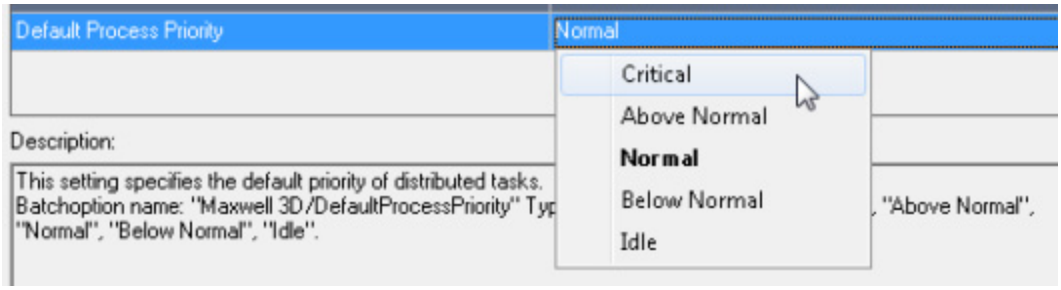
Why the GPU May Not Be Used:

- The time estimation is based on the entire matrix, not just the size. Having larger matrices doesn't necessarily mean GPU will be used.
- The bandwidth does not necessarily determine usage either. It is possible to have two matrices of exactly the same size and bandwidth, one favoring GPU and the other CPU.
- If GPU is an older model while CPU is newer, the GPU may not be used.
- If CPU has a high clock rate, the GPU may not be used.
- If CPU has many cores, the GPU may not be used.

Multiple GPUs

When there are multiple GPU cards in the same machine, the first 3D eddy solver process locks the first GPU card available (that is, one that is not already locked or used by other processes, not used for display, etc.). The second process locks the second available card, and so on. If all cards are locked, the solver uses the CPU only.

9. Optionally, you can select one of the following from the Default Process Priority pull-down list:



   - Critical (highest) Priority (not recommended)
   - Above Normal Priority (not recommended)
   - Normal Priority
   - Below Normal Priority
   - Idle (lowest) Priority

   To edit configurations, see Editing Distributed Machine Configurations.

Related Topics

High Performance Computing (HPC) Integration

"Using GPU Acceleration for Maxwell 3D Solvers " on the next page

Specifying the Remote Spawn Command as RSH or SSH (Linux)

An important step in using a high performance cluster is setting up authentication across machines in such a way that the machines can be accessed without a password. By default Maxwell 2D and Maxwell 3D use SSH authentication on Linux to spawn commands on the remote machines but also supports RSH. The selection of which to use is made on the **Options** tab of the Tools>Options>HPC and Analysis dialog.

SSH

You will need to set up passwordless access to useMaxwell on a Linux cluster with SSH or RSH. In general, for SSH, this is accomplished as follows:

1. Verify that you have working SSH servers and clients on your machines.
2. Verify that the server will accept passwordless logins. You may need to edit the /etc/ssh/ssh_d file to allow RSAAuthentication and PubkeyAuthentication.
3. Generating keys on the client system using the ssh-keygen program. Do not use a passphrase so that you can access the machine without a password.
4. Copy the public key generated in step 1 from the ~/.ssh directory to the server. The easiest way to transfer the keys is to use the ssh-copy-id program. Alternately, you can use any file

transfer utility. If the server already has a list of existing keys for other clients add the new public key to the list.

5. Test the connection. Login to the client machine using the username that you used to create the identity keys. Open a new shell terminal and attempt to open an SSH login session. For example type: ssh 192.168.0.4 (where the IP address is the address of the machine you are attempting to connect to). The server should allow you to login without requesting a password.

RSH

If you choose to use RSH you will need to make sure RSH is installed on all the machines and set the machines up so that you are not prompted for a password. There are different ways to set up password-less RSH so be sure to see the documentation for your machines and network for detailed instructions.

Machine access using RSH without a password is often set up by editing the /etc/hosts.equiv file and adding entries for the hosts you would like to use without a password. This file lists hosts and users that are granted "trusted" access to the system.

If you look at the contents of the `/etc/hosts.equiv` file you should have something similar to the following:

```
job1.n1.com
job2.n1.com
job3.n1.com
```

The machines job1, job2 and job3 can connect without a password. You may also need to verify that the files /etc/hosts.allow and /etc/hosts.deny are empty. See your local documentation for detailed instructions and troubleshooting suggestions.

Related Topics

"Setting HPC and Analysis Options for Maxwell and RMxprt Designs" on page 4-54

Using GPU Acceleration for Maxwell 3D Solvers

Maxwell 3D solvers can be accelerated by using an NVIDIA® GPU accelerator. Numerical factorization of symmetric matrices is supported for GPU acceleration.

You can also enable distributed GPUs for Eddy Current Frequency Sweeps.

Hardware Requirements

GPU acceleration for the Maxwell 3D Eddy Current solver has been developed for NVIDIA cards and is officially supported with the Tesla® series. NVIDIA Tesla cards are recommended for the best performance when using several cards on one machine to solve using HPC. The following cards are supported:

- NVIDIA Tesla K20c (both workstation and server)
- NVIDIA Tesla K40c (both workstation and server)
- NVIDIA Tesla K80 (server with GPU cooling solution only, 2 GPUs consume 2 Ansys HPC licenses)
- NVIDIA Quadro K5000, K5200, M6000
- NVIDIA Quadro K6000 (both workstation and server, will be slow in graphic processing when used for GPU acceleration)

Note	NVIDIA Tesla M2090 (Not supported in Release 17), a previous generation (code Fermi) GPU card, doesn't work for Workstation since it has no fan for active cooling, but rather needs a server with GPU cooling solution (passive cooling) similar to the NVIDIA Tesla K80.

To obtain the best performance, the GPU used for running simulation jobs should not be attached to any display. Only GPU cards with CUDA® Compute Compatibility 2.0 and above should be used. To improve the speedup of visualization, you should install the GPU card on a system with PCI-E 3.0 slots. A mixture of interface cards with lower PCI-E versions may result in the data not being transferred from GPU to CPU at the highest speed.

Setup for Windows

1. After you install GPU cards and Nvidia graphics drivers, you should be able to find the cards in Windows Display Manager**.**
2. Run `nvidia-smi.exe` at C:\Program Files\NVIDIA Corporation\NVSMI to check if GPU cards are installed successfully. (The executable `nvidia-smi.exe` should be available after the display driver is installed.)
3. To further set up the configuration of GPU cards, open a command window as an administrator.
4. To improve the performance of GPU acceleration, it is recommended that you turn off the Error Correction Code (ECC) support using the `-e 0` option of nvidia-smi.

   ```
   nvidia-smi.exe -e 0
   ```

   New ECC settings become effective only after system reboot.
5. (Optional) For remote execution of GPU accelerated jobs (e.g., through Windows Remote Desktop Connection or RSM options), it is necessary to turn on the Tesla Compute Cluster (TCC) mode using the `-dm 1` option of nvidia-smi.

   ```
   nvidia-smi-exe -dm 1
   ```

   New TCC settings become effective only after system reboot.

Note	• The above step is unnecessary if you run Maxwell 3D Eddy Current from a local machine. • Only Tesla cards support TCC. • You cannot run GPU accelerated jobs on remote GeForce and Quadro cards.

Setup for Linux

1. After you install an NVIDIA GPU card and graphics drivers, you should be able to find the card using the command:

   ```
   /sbin/lspci | grep -i nvidia
   ```

   You can also use the following command to check if a GPU card can be recognized by the system:

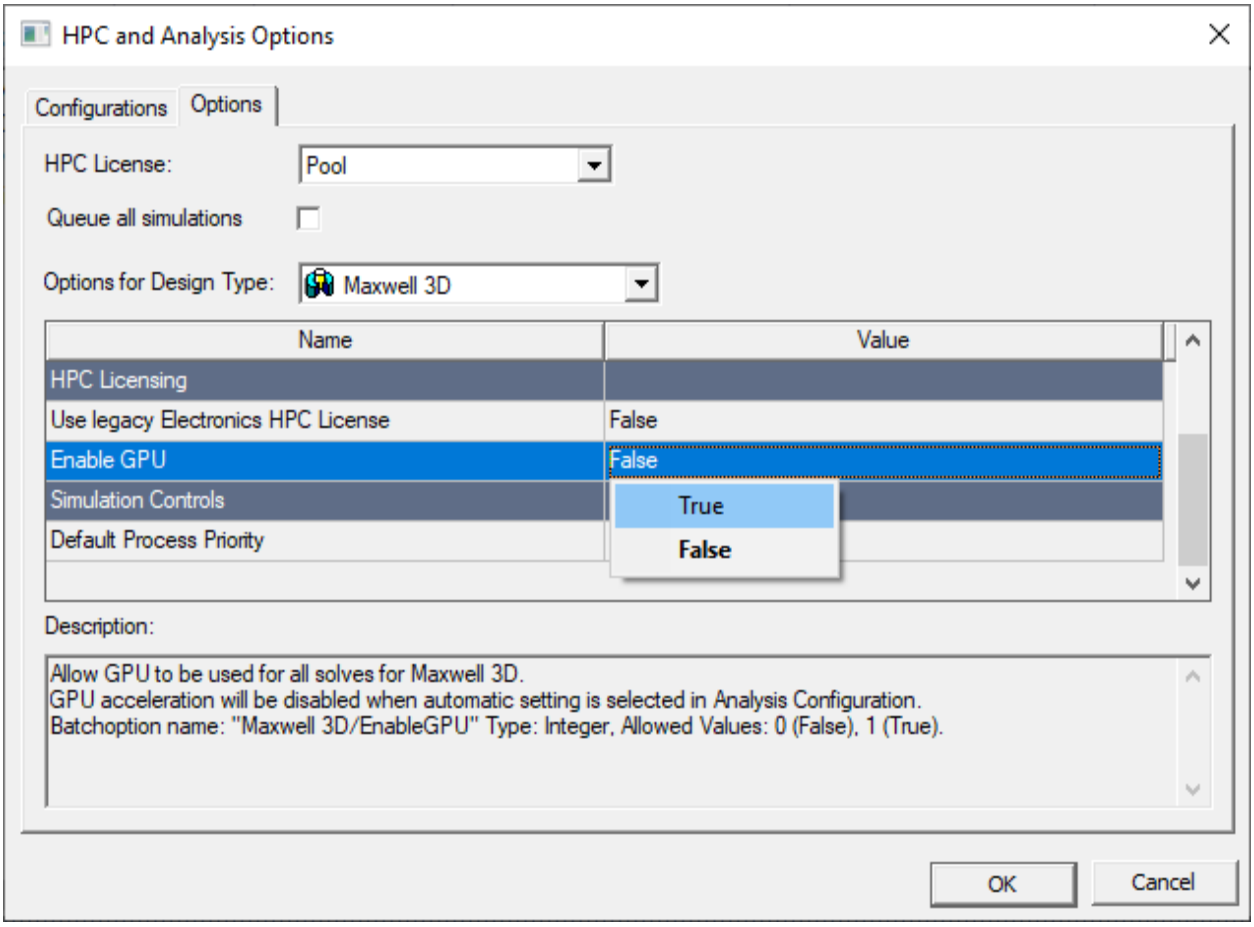
   ```
   /usr/bin/nvidia-smi
   ```

2. (Optional) The setting of GPUs to disable ECC (for performance), enable TCC (for remote execution), and enable Exclusive_Process (for GPU-distributed) are similar to Windows. You need the administrative rights to make such changes.

   ```
   sudo nvidia-smi -e 0
   sudo nvidia-smi -dm 1
   sudo nvidia-smi -c 3
   ```

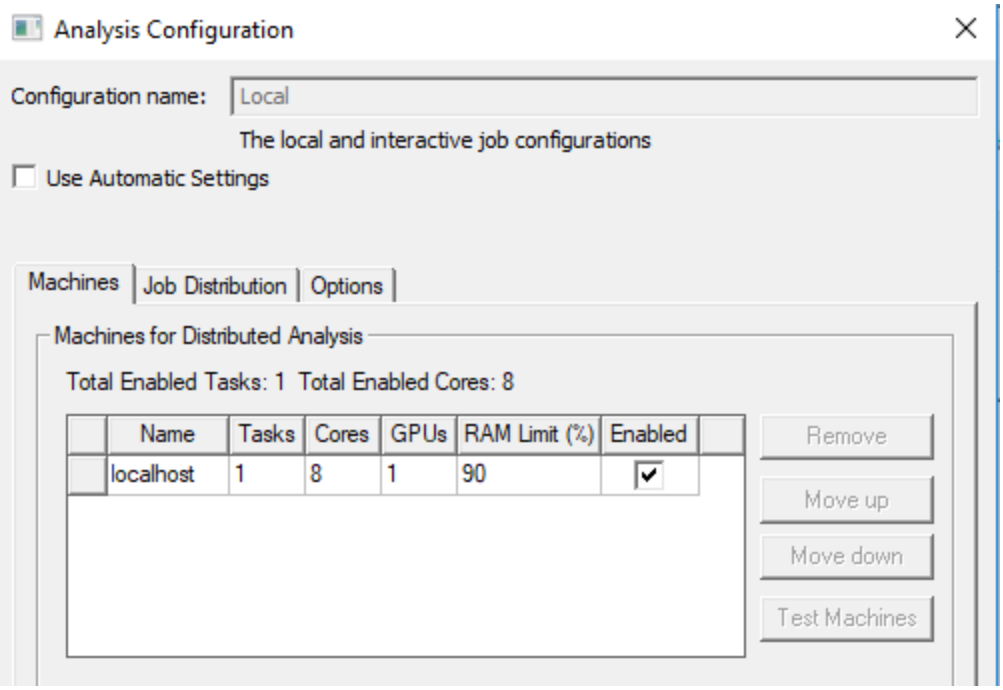
Enabling GPUs for Maxwell 3D solvers

1. Go to **Tools>Options>HPC and Analysis Options**.
2. In the Options tab, set the **Enable GPU** option to **True**.



Setting Up Non-distributed GPU for Maxwell 3D Solvers

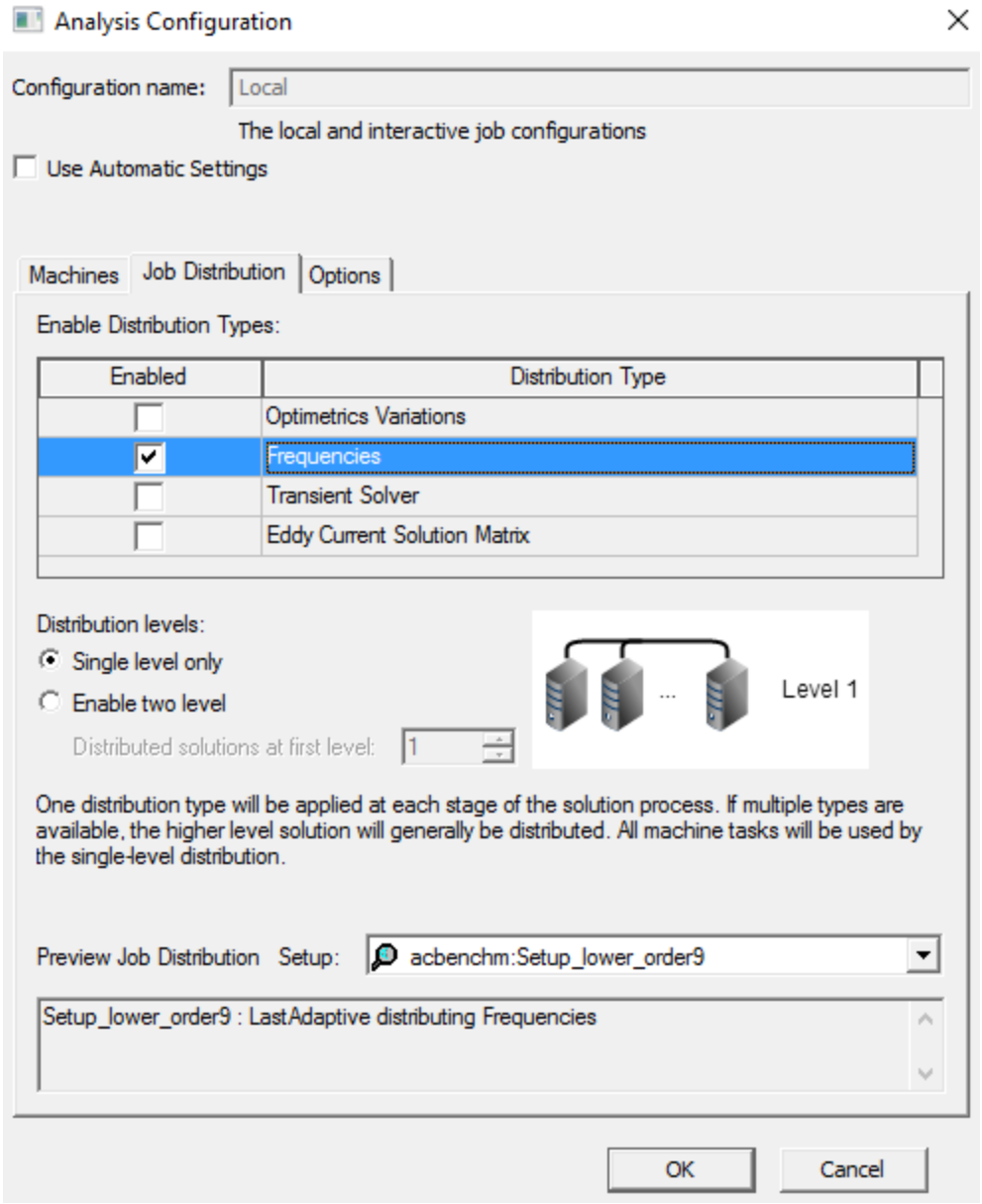
In the **Analysis Configuration** dialog box **Machines** tab, in the **Machines for Distributed Analysis** section, set **GPUs** to 1.



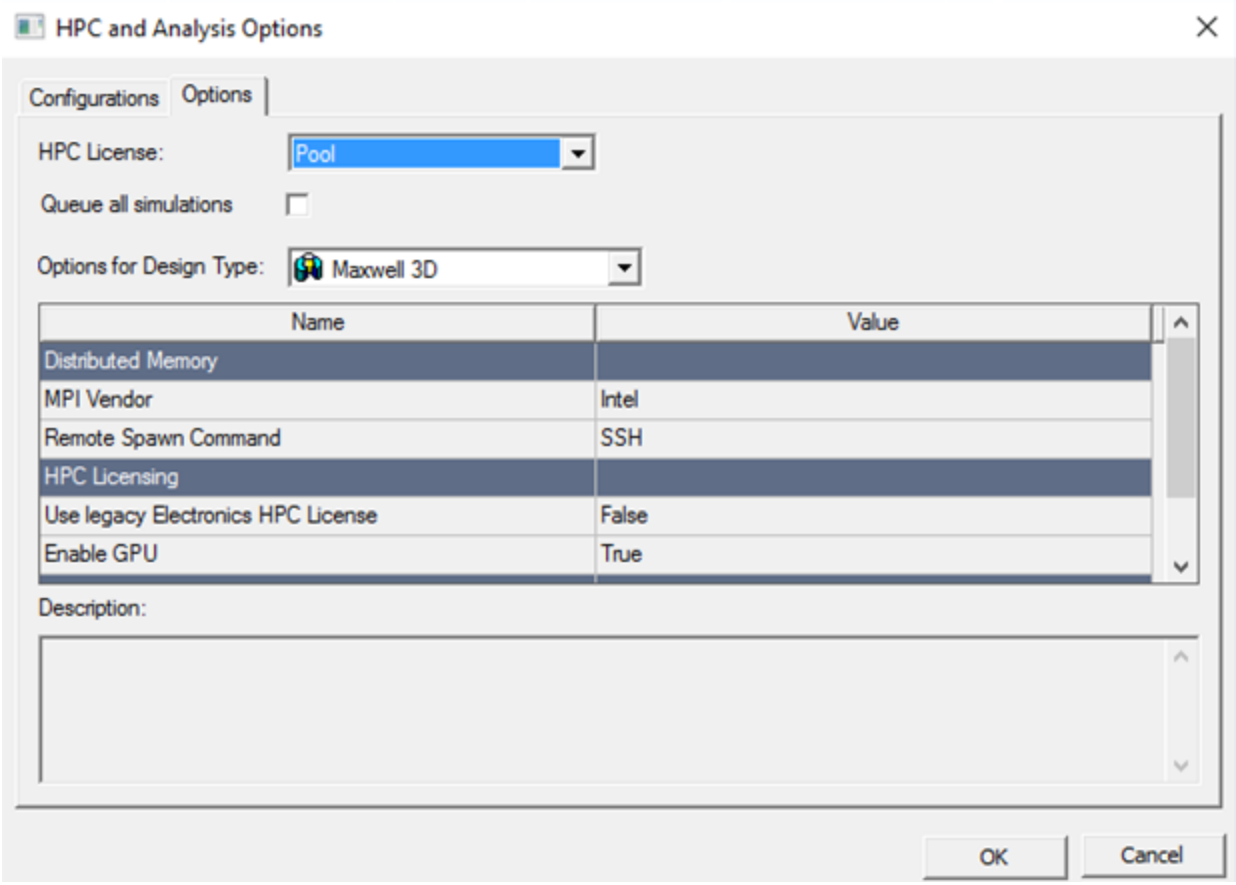
Enabling Distributed GPUs for Eddy Current Frequency Sweep

To set up distributed GPUs for eddy current frequency sweeps:

1. Go to **Tools>Options>HPC and Analysis Options** and click **Edit** in the dialog box, or select the **Simulation** tab of the ribbon, and click **Analysis Config** to edit Analysis Configurations.
2. Uncheck **Use Automatic Settings**, select the **Job DIstribution** tab, then select **Frequencies**.

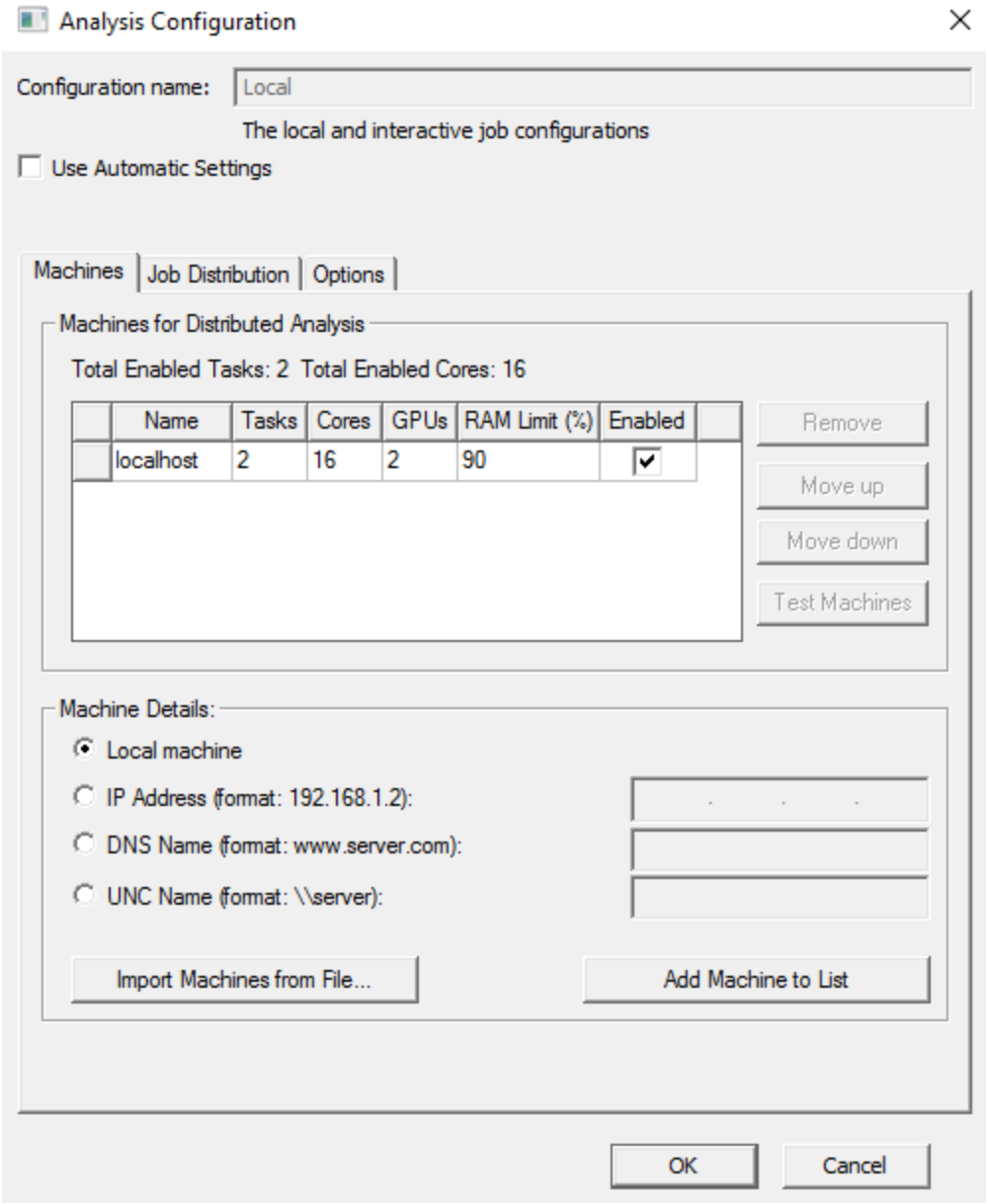


3. Go to **Tools>Options>HPC and Analysis Options**, select the **Options** tab, and set **Enable GPU** to **True**.



4. Set up the Machines for distributed GPU usage.

   For example, the following figure shows two simulation jobs set to run on the local host; and up to two GPUs can be used for acceleration depending on the availability of licenses and GPU cards.

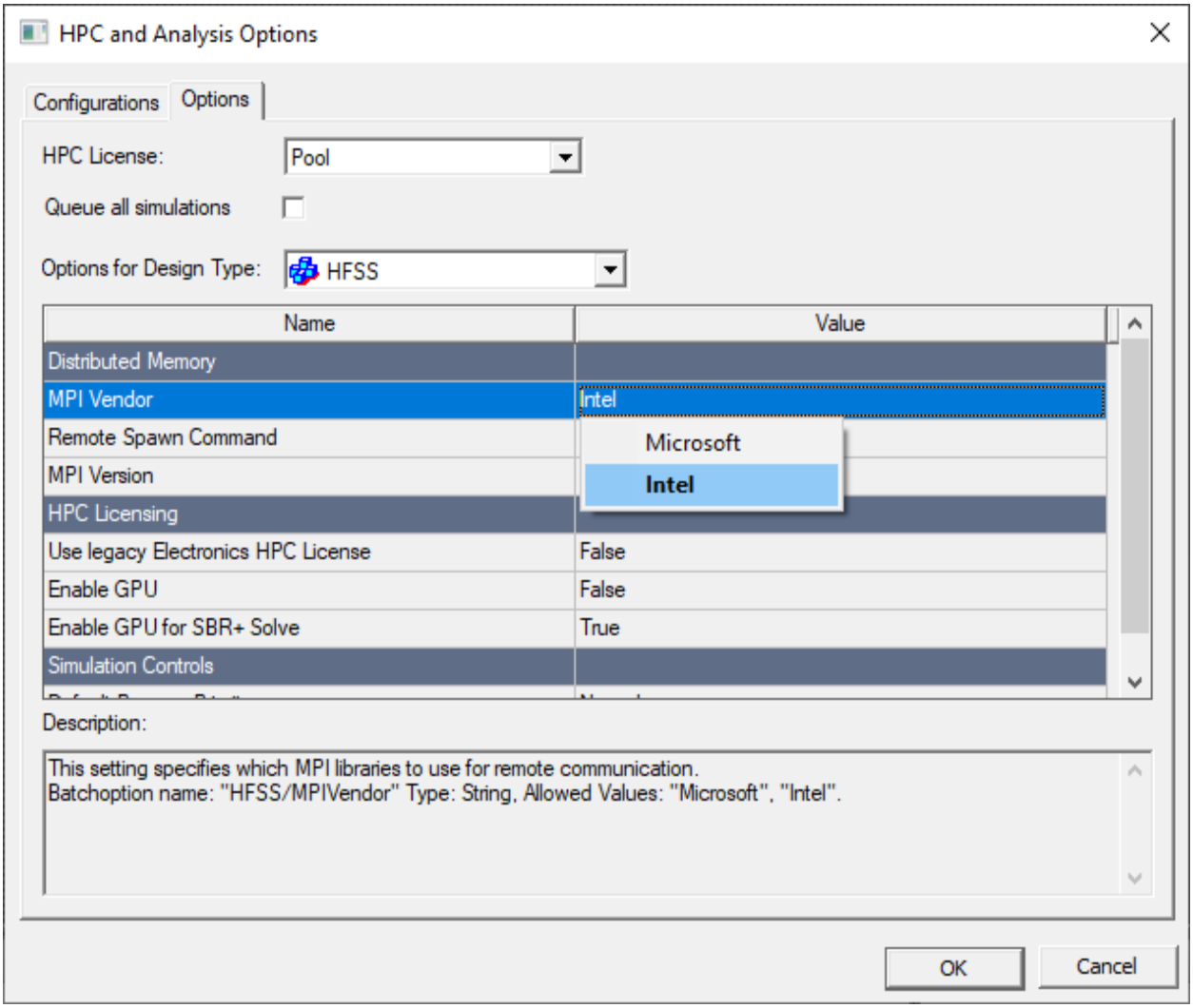


Related Topics

"Setting HPC and Analysis Options for Maxwell and RMxprt Designs" on page 4-54

Select the MPI Vendor

Aafter installing MPI on your machine from a particular vendor such as Microsoft or Intel, you need to set which type of MPI you are using: Microsoft or Intel. Go to the **Options** tab of the **HPC and Analysis Options** dialog box, and select the design type and the MPI Vendor type.



Maxwell 2D/3D Options

Select the **Options** page of interest:

Maxwell 3D Options

Maxwell 2D Options

Setting Maxwell 3D Options

To set Maxwell 3D options:

1. Click **Tools>Options>General Options** to open the **Options** dialog box, displaying a hierarchical list of options. Click the + next to **Maxwell 3D** to show the following choices: **General**, **Solution Type**, **Material Threshold**, **Boundary Assignment**.
2. On the **General** panel, select or clear the following check boxes:

- **Save before solving**

Note	When you enable the **Save before solving** setting, the project is saved only if it has been modified since its last save.

- **Save Optimetrics field solutions**
- **Apply variation deletions immediately**

3. On the **Solution Type** panel:
   - To change the default solution type when you initially insert a design, select one of the following from the **Default solution type** pull-down list: **Electrostatic**, **Magnetostatic**, **Eddy Current**, **Transient**, **DC Conduction**, **ElectroDCConduction**, **Electric Transient**, **TransientAPhiFormulation**.
4. On the **Material Threshold** panel, enter the **Default perfect conductor** and **Default insulator/conductor** values in siemens/m.

Note	Setting the material threshold affects the default setting for the current and all future projects/designs. To change the material threshold for the current design only, use the **Maxwell>Design Settings** command and change the material thresholds on the **Set Material Thresholds** tab.

5. On the **Boundary Assignment**panel, select or clear the following check boxes:
6. **Use Wizards for data input when creating new boundaries**

   When this is checked, the creation of boundaries and excitations use Wizards to guide you through the process. When this is not checked, the creation of boundaries and excitations displays a Properties dialog with tabs for different kinds of information.
   - **Duplicate boundaries/mesh operations with geometry**

     When this is checked, you can duplicate a boundary or excitation when its geometry is pasted or duplicated. See Duplicating Boundaries and Excitations.
   - **Visualize boundaries on geometry**

     When checked, enables visualization of boundaries on geometry.

Related Topics

Setting the Material Threshold

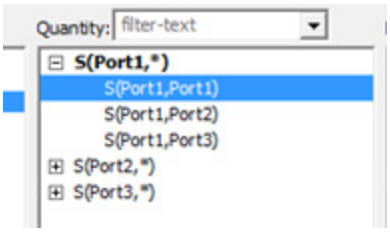
Report Setup Options

To set up general options for Reports

1. Click **Tools>Options>General Options** to open the **Options** dialog box, displaying a hierarchical list of options. Click the + next to **Reporter** to expand the list and select **Report Setup**.
2. In the **Report Setup** panel, set the **Maximum number of significant digits to display in the reporter when displaying numeric quantities**.
3. In the **Drag and Drop** section, select either **Drag Item Data**, or **Drag Item Definition**.

4. The **Quantity selection** value specifies the matrix size for using a tree display for matrix quantities. This is helpful when dealing with larger matrices. The default is 50. When the number of matrix elements is larger than the number, the Quantities field uses a tree structure to divide matrix quantities into groups by their first element name. The initial display shows groups, without initially listing group members.

   A folder Node is not selectable. A click on a folder node toggles (Expand or collapse) the node. When any of a folder's child nodes is selected it becomes bolded.



   Mouse Click on Quantity Node (Tree leaf node). Shift and Ctrl key only apply to multiple selection dialogs:

   - Without Shift and Ctrl key – Select the quantity and unselect all previous selected quantities.
   - Only with Ctrl key down – Toggle the selection of the quantity. No effect on other selected quantities.
   - Only with Shift key down – Do range selection, deselect any selected quantity that is outside of the range.
   - Both Shift and Ctrl key down – Do range selection, but don't deselect any selected quantity.
   - Ctrl+a – Select all quantities in a multiple selections dialog.

   Range selection: Select quantity nodes between the last mouse clicked quantity node and the newly clicked-on quantity node. Folder nodes in between won't be selected but their children will be selected. So those folder nodes will be in a bolded state.

5. Click **OK**.

Setting Maxwell 2D Options

To set Maxwell 2D options:

1. Click **Tools>Options>General Options** to open the **Options** dialog box, displaying a hierarchical list of options. Click the + next to **Maxwell 2D** to show the following choices: **General**, **Solution Type**, **Material Threshold**, **Boundary Assignment**.
2. On the **General** panel, select or clear the following check boxes:
   - **Save before solving**

Note	When you enable the **Save before solving** setting, the project is only saved if it has been modified since its last save.

   - **Save Optimetrics field solutions**

- **Apply variation deletions immediately**
- **Generate model history when importing legacy 2D projects**

3. On the **Solution Type** panel:
   - To change the default solution type when you initially insert a design, select one of the following from the **Default solution type** pull-down list: **Magnetostatic**, **Eddy Current**, **Transient**, **Electrostatic**, **DC Conduction**, **AC Conduction**.
   - To change the default geometry mode when you initially insert a project, select one of the following from the **Default geometry mode** pull-down list: **XY**, **about Z**.
4. On the **Material Threshold** panel, enter the **Default perfect conductor** and **Default insulator/conductor** values in siemens/m.

Note	Setting the material threshold affects the default setting for the current and all future projects/designs. To change the material threshold for the current design only, use the **Maxwell>Design Settings** command and change the material thresholds on the **Set Material Thresholds** tab.

5. On the **Boundary Assignment** panel, select or clear the following check boxes:
   - **Use Wizards for data input when creating new boundaries**

     When this is checked, the creation of boundaries and excitations use Wizards to guide you through the process. When this is not checked, the creation of boundaries and excitations displays a Properties dialog with tabs for different kinds of information.
   - **Duplicate boundaries/mesh operations with geometry**

     When this is checked, you can duplicate a boundary or excitation when its geometry is pasted or duplicated. See Duplicating Boundaries and Excitations.

Related Topics

Setting the Material Threshold

Setting RMxprt Options

To specify settings for RMxprt options:

1. Click **Tools>Options>General Options** to open the **Options** dialog box, displaying a hierarchical list of options. Click the + next to **RMxprt** to show the following choices: **General**, **Machine Type**, **Threshold**
2. In the **General** options panel, select or clear the following check boxes:
   - **Save before solving**

Note	When you enable the **Save before solving** setting, the project is only saved if it has been modified since its last save.

   - **Apply variation deletions immediately**
3. On the **Machine Type** panel, to change the default machine type when you initially insert a design, select one of the following from the **Default machine type** pull-down list:

- Three Phase Induction Motor
- Single Phase Induction Motor
- Three Phase Synchronous Machine
- Brushless Permanent-Magnet DC Motor
- Adjust-Speed Synchronous Machine
- Permanent-Magnet DC Motor
- Switched Reluctance Motor
- Line-Start PM Synchronous Motor
- Universal Motor
- DC Machine
- Claw-Pole Synchronous Machine
- Three Phase Non-Salient Synchronous Machine
- Generic Rotating Machines

4. On the **Threshold** panel, enter the **Default conductivity** and **Default permeability** values in siemens/m.

Note	Setting the material thresholds under **Tools>Options** impacts the default setting for the current and all future projects/designs. To change the material threshold for the current design only, use the **RMxprt>Design Settings** command and change the material thresholds on the **Set Material Thresholds** tab.

5. Click **OK** to close the dialog box.

Setting Report2D Options

To set Report2D options in Maxwell:

1. Click **Tools>Options>General Options** to open the **Options** dialog box, displaying a hierarchical list of options. Click the + next to **Reporter** to expand the list and select **Report2D** to display the following list:
   - Curve
   - Axis
   - Grid
   - Header
   - Note
   - Legend
   - Marker
   - Marker Table
   - X-Y Markers
   - Stacked
   - Digital
   - General
   - Table

For properties controlled by check boxes, you can set values for all curves by clicking the column header cell that contains the property title. Right-clicking on a text field cell displays a context menu that lets you cut, copy and paste values. Right-clicking on a menu cell displays a context menu that lets you copy and paste entire rows.

You can use a **Restore Defaults** button.

2. Click each list item, and make the desired selections.
3. Click **OK**.

Report 2D Options: Curve

These options are set on the **Curve** panel under **Report2D** in the **Options** dialog.

- **Line style** – select the options from the drop down menu. The options are Solid, Dot, Dash, and Dot dash.
- **Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Width** – set the line width by editing the real value in the text field.
- **Arrows** – use the check box to use arrows on the curve ends.
- **Symbol** – use the check box to have symbols mark the locations of data points on the curve.
- **Sym Freq** – set the symbol frequency by editing the integer value in the text field.
- **Sym Style** – select the symbol to display for the designated data points. The sym style can be box, circle, vertical ellipse, horizontal ellipse, vertical up triangle, vertical down triangle, horizontal left triangle, horizontal right triangle.
- **Fill Sym** – use the check box to set the symbol display as a solid or as hollow.
- **Sym Color** – set the color for the symbol by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.

Report2D Options: Axis

These options are set on the **Axis** panel under **Report2D** in the **Options** dialog.

- **Axis Name** – this describes the axis to which the following options refer.
- **Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Font color** – set the font color of the axis by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Edit Font** – click the cell to display the Edit Text Font dialog box. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. **OK** the selections to apply the font edits and to close the dialog.
- **Font Description** – displays the chosen font properties.

Report2D Options: Grid

These options are set on the **Grid** panel under **Report2D** in the **Options** dialog.

- **Grid Name** – lists the name or letter of the grid. Not editable.
- **Line Style** – select the options from the drop down menu. The options are Solid, Dot, Dash, and Dot dash.
- **Line Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.

Report2D Options: Header

These options are set on the **Header** panel under **Report2D** in the **Options** dialog.. For the Title and SubTitle, you can independently specify the following:

- **Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Font** – click the cell to display the Edit Text Font dialog box. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. **OK** the selections to apply the font edits and to close the dialog.

Report2D Options: Note

These options are set on the **Note** panel under **Report2D** in the **Options** dialog.

- **Note Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Note Font** – click the cell to display the Edit Text Font dialog box. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. **OK** the selections to apply the font edits and to close the dialog.
- **Background Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Background Visibility** – use the check box to toggle the background for the note on or off.
- **Border Line Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Border Visibility** – use the check box to toggle the visibility of the note border.
- **Border Line Width** – set the line width by editing the real value in the text field.

Report2D Options: Legend

These options are set on the **Legend** panel under **Report2D** in the **Options** dialog.

- **Show Trace Name** – use the check box to toggle the visibility of the trace name.
- **Show Solution Name** – use the check box to toggle the visibility of the solution name.
- **Show Variation Key** – use the check box to toggle the visibility of the variation key.

- **Text Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Text Font** – click the cell to display the Edit Text Font dialog box. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. **OK** the selections to apply the font edits and to close the dialog.
- **Background Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Border Line Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Border Line Width** – set the line width by editing the real value in the text field.
- **Grid Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.

Report2D Options: Marker

These options are set on the **Marker** panel under **Report2D** in the **Options** dialog.

- **Marker Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Marker Font** – click the cell to display the Edit Text Font dialog box. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. **OK** the selections to apply the font edits and to close the dialog.
- **Arrow Direction** -- set the arrow direction by choosing Up, Down, Left, or Right from the drop-down menu.

Related Topics

Modifying Markers on Point Plots

Report2D Options: Marker Table

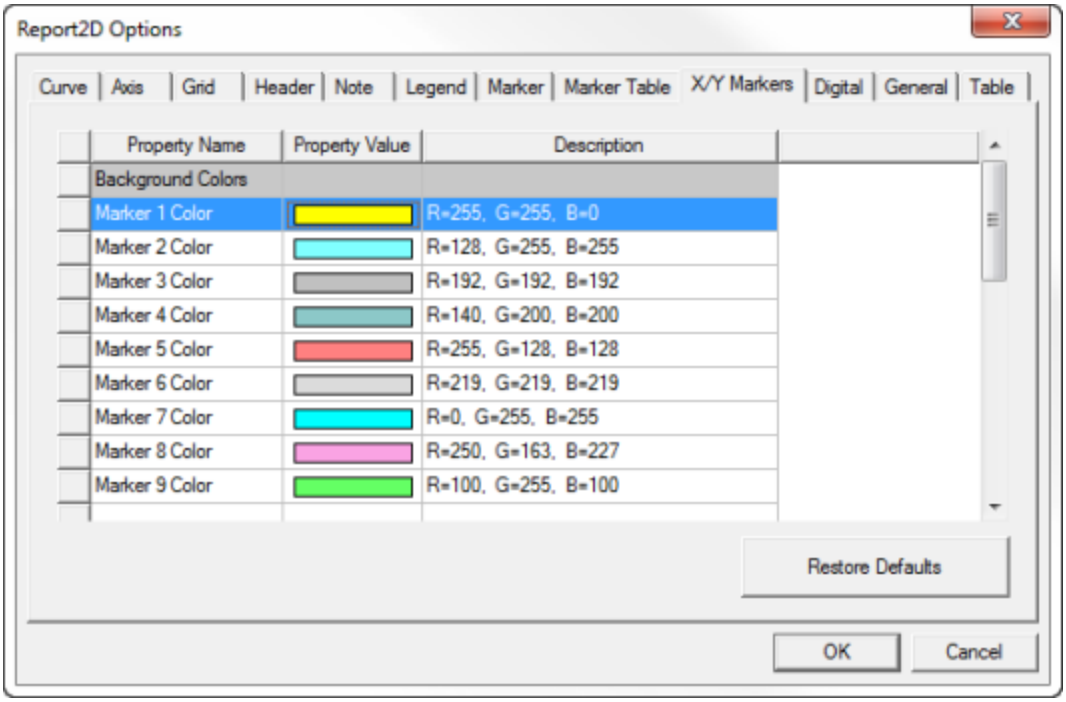
These options are set on the **Marker Table** panel under **Report2D** in the **Options** dialog.

- **Text Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Text Font** – click the cell to display the Edit Text Font dialog box. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. **OK** the selections to apply the font edits and to close the dialog.
- **Background Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Border Line Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Border Line Width** – set the line width by editing the real value in the text field.
- **Grid Color** – set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Grid Line Width** – set the line width by editing the real value in the text field.

Report 2D Options: X-Y Markers

These options are set on the **X-Y Markers** panel under **Report2D** in the **Options** dialog.

You can set background colors for Markers 1 through 10 by clicking the current color to open a color selection dialog box.



In addition to marker colors, properties you can set include:

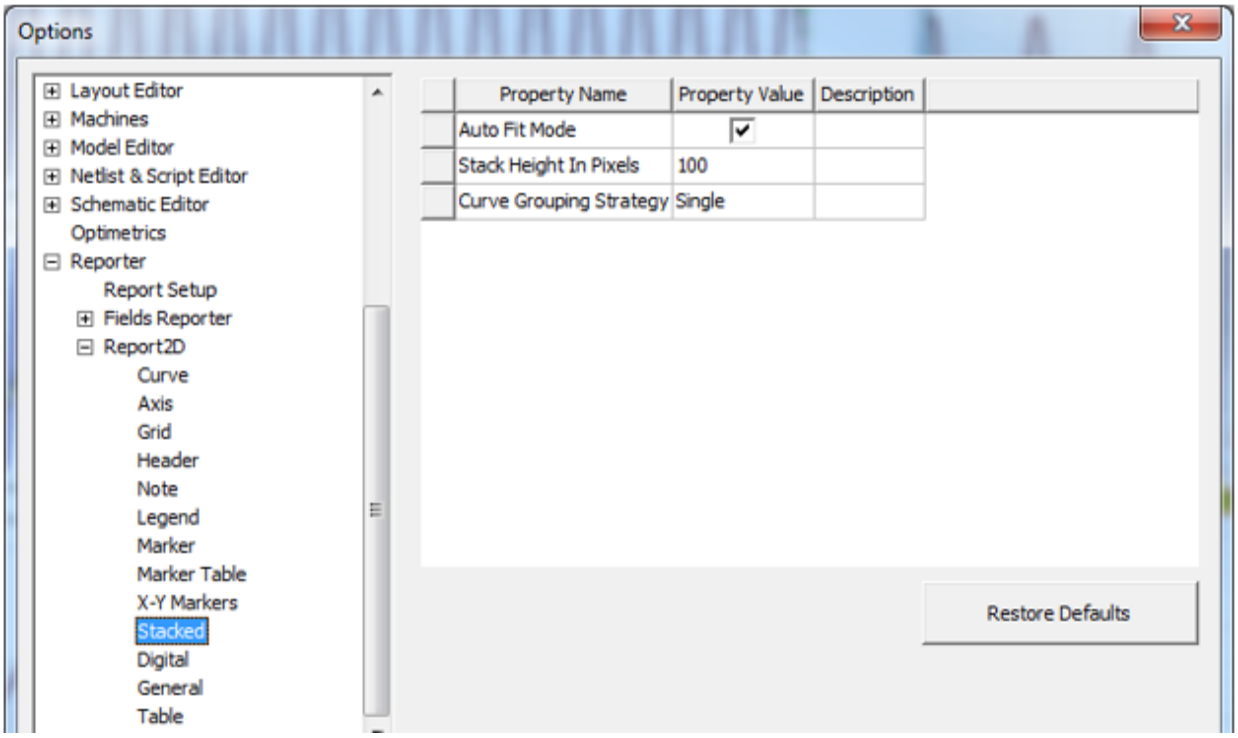
- On-screen intersection
- Marker Font
- Text color
- Line color
- Line style
- Line width
- Whether to Show Name
- Whether to Snap to Vertex

Inter marker delta properties include:

- Whether to show Delta
- Delta font
- Delta text color
- Line color
- Line style
- Line width

Report 2D Options: Stacked

These options are set on the **Stacked** panel under **Report2D** in the **Options** dialog.



- Auto Fit Mode – **On** or **Off**
- Stack Height in Pixels – sets the default stack height
- Curve Grouping Strategy -- can be **Single**, **By Trace**, or **By Units**. **Single** means that a new stacked plot shows a single curve per stack. **By Trace** means that all curves are grouped by their trace. **By Units** means that all curves are grouped by their unit type.

If you change the **Curve Grouping Strategy** set by default, existing stacked plots remain unaffected. The new default will apply only to new stacked plots.

When a project containing stacked plots saved in versions before 19.0 is opened in version 19.0, **Curve Grouping Strategy** will default to **Single** and the stacked plots remain unaffected.

Related Topics

Creating a 2D Rectangular Stacked Plot

Report2D Options: Digital

These options are set on the **Digital** panel under **Report2D** in the **Options** dialog.

- Digital Literal Foreground color
- Whether to Expand Arrays/Records

Digital Stack Height in Pixels can be set for the following:

- Analog
- Digital
- Enum
- Event
- Literal

Report2D Options: General

These options are set on the **General** panel under **Report2D** in the **Options** dialog.

- **Background Color** – set the color by clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Plot Area Color** – set the color by clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Highlight Color** – set the color by clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Accumulate Depth** – enter a value; the default is 4..
- **Enable Y Axis Stripes** – select the check box to add stripes to the Y axis; deselect to remove stripes.
- **Auto Scale Fonts** – on by default and when enabled scales text in plots and color key (contour plot, field plots in 3D modeler) for high resolution screens.

Curve Tooltip

- **Show Trace Name** – select the check box to display trace names in the tooltip when hovering the cursor over a curve.
- **Show Variation Key** – select the check box to display the variation key in the tooltip when hovering the cursor over a curve.
- **Show Solution Name** – select the check box to display the solution name in the tooltip when hovering the cursor over a curve.

Clipboard Option

- **Capture Aspect Size Ratio** – this can be As Shown or Full Screen.
- **Capture Background Color** – this can be As Shown or White.

Format

- **Field Width** – set the number of digits to display by editing the real value field.
- **Precision** – set the precision for marker placement by editing the real value field.
- **Use Scientific Notation** – use the check box to toggle scientific notation on or off.

Report2D Options: Table

These options are set on the **Table** panel under **Report2D** in the **Options** dialog.

- **Rows Per Page** – set to 2500 by default.
- **Text Color** – set the color by clicking to display the Set color dialog box. Select a default or custom color and click **OK**.

- **Text Font** – click the cell to display the Edit Text Font dialog box. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. **OK** the selections to apply the font edits and to close the dialog.
- **Border Width** – set to 2 by default.
- **Border Color** – set the color by clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Grid Width** – set to 1 by default.
- **Grid Color** – set the color by clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Background Color** – set the color by clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Page Link Color** – set the color by clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Arrow Color** – set the color by clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Header Row** – use the following properties to set the header row format:
  a. **Text Color** – set the color by clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
  b. **Text Font** – click the cell to display the Edit Text Font dialog box. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. **OK** the selections to apply the font edits and to close the dialog.
  c. **Background Color** – set the color by clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
- **Format** – use the following properties to set the format:
  a. **Field Width** – set the table field width by editing the real value in the text field.
  b. **Precision** – set the table precision by editing the real value in the text field.
  c. **Use Scientific Notation** – use the check box to toggle scientific notation on or off.
- **Copy to Clipboard** – use the following check boxes to toggle the following properties for table copy operations:
  a. **With Header**
  b. **With Tab Separator**

Setting 3D Modeler Options

To set 3D modeler options:

1. Click **Tools>Options>General Options** to open the **Options** dialog box, displaying a hierarchical list of options.
2. Click the + next to **3D Modeler** to display the following choices:
   - Drawing
   - Operation
   - Snap
   - Display

- Group
- SpaceClaim Link
- Advanced

2. Click each option, and make the desired selections.
3. Click **OK**.

3D Modeler Options: Drawing

These options are set on the **Drawing** panel under **3D Modeler** in the **Options** dialog.

1. The **Drawing Data Entry Mode** section controls whether you draw new objects directly via the mouse (**Point**), or whether a **Properties** dialog opens for you to enter dimensions for the object. The **Dialog** mode drawing feature works with the equation based line, and all two and three dimensional objects. You can also use **F3** to switch to Point mode and **F4** for dialog mode.
2. In the **Relative Coordinate System Creation Mode** radio buttons control whether the default uses **Axis/Position** or **Euler Angle** to create a new relative coordinate system. You can use the F3/F4 keys to switch between the modes.
   - For the **Axis/Position** setting, **Modeler> Create> Relative CS>[Offset | Rotated | Offet and Rotated]** enters **Draw** mode, displaying a point selection cursor and the **Measure Data** dialog box. Also the **Status** bar fields for setting X, Y and Z values, as **Coordinate System** settings are active.
   - For the **Euler Angle** setting, **Modeler>Create>Relative CS>[Offset | Rotated | Offset and Rotated]** displays a dialog in which you type in the needed values and select units.
3. In the **Polyline Creation** section, select or clear the **Automatically cover closed polylines** check box.

   By default, surface objects created with the **Polyline** command will be created with a cover so that they become sheet objects. You can choose to leave the polyline as an uncovered object to perform further operations prior to creating a sheet object
   - If checked, closed polylines become sheet objects, and are listed as such in the History tree.
   - If not checked, closed polylines are listed under lines in the History tree.
4. Select or clear the **Show measures dialog during drawing** check box. This specifies whether a Measure dialog appears on the creation of a new primitive. The dialog shows the coordinates of the current cursor position.
5. To have a Properties dialog display whenever you create a new object in the modeling window, check the box for **Edit properties of new primitives**.

3D Modeler Options: Operation

These options are set on the **Operation** panel under **3D Modeler** in the **Options** dialog.

1. By default, the modeler will delete tool objects when performing tasks such as Uniting, Subtracting, or Intersecting objects. The user may specify that the modeler should make a copy (clone) of the tool object prior to the operation, allowing the object to remain available for subsequent operations. To specify when to clone tool objects, select or clear the following check boxes in the **Clone** section:
   - **Clone tool objects before uniting**
   - **Clone tool objects before subtracting**
   - **Clone tool options before intersecting**
   - **Clone tool objects before imprinting**
   - **Clone tool objects before projecting**
2. In the **Coordinate System** section, select or clear the **Automatically switch to face coordinate system** check box.

   By default, the modeler operates within the user selected coordinate system. If this option is enabled, you can select a face and when a new object creation is started, the modeler first creates a face coordinate system consistent with the selected face and the new object is created within the face coordinate system. With this selection, unchecked, you must manually create a Face Coordinate System before creating an object related to it.
3. For the **Model Edit** section:
   - If **Delete invalid objects created during split operation** is checked, the modeler deletes invalid objects created during split operation. If not checked, invalid objects can be created. Validation issues warnings.
   - Select or clear **Automatically imprint wrapped sheets**.
4. In the **Model Save** area, you can enable/disable being prompted for model history cleanup. You can also specify the number of actions on a part that will cause a prompt. The default value is 50.

3D Modeler Options: Snap

These options are set on the **Snap** panel under **3D Modeler** in the **Options** dialog.

1. To specify snap settings, select or clear the following check boxes in the **Snap Mode** section:
   - Grid
   - Vertex
   - Edge Center
   - Face Center
   - Quadrant
   - Arc Center
2. Enter how near the mouse needs to be to click a grid item in the **Mouse Sensitivity** box, in pixels.

3D Modeler Options: Display

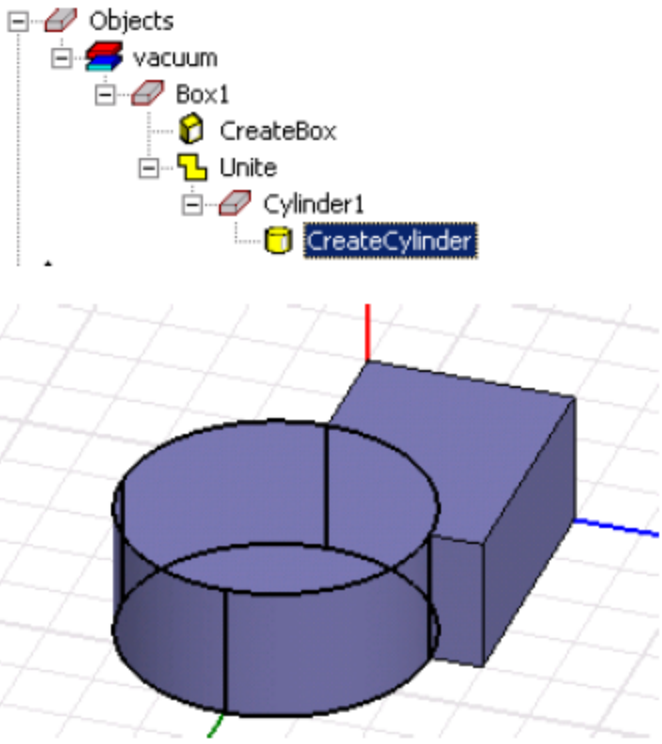
Under **3D Modeler** in the **Options** dialog box, the **Display** options settings are in three sub-groups:

- **General**
- **Rendering**
- **History Tree**

General

1. Under **Operations History Visualization**, select or clear the **Visualize history of objects** check box.

   This option lets you view an outline of each part that comprises an object when the given part is selected in the model history tree. This can help you visualize an object that has been merged with another object. A change to the option takes effect only when you restart Ansys Electronics Desktop. Clearing this selection removes visualization of objects that are part of the model history. For large models, this is faster and uses less memory. The following figure shows an example history tree with an object selected and the outline view of that object in the Main window.



2. Select or clear **Show coordinate system of selected operations**.

3. Under Selection:
    a. Select or clear the **Show orientation of selected objects** check box.
    b. Select or clear the **Highlight selection dynamically** check box. This option causes objects or faces (depending on the selection mode) to be highlighted when you pass the mouse pointer over them. You may want to turn this off for complicated models as it can reduce responsiveness. For large models, the dynamic highlighting option is automatically ignored, and a message is displayed.
4. Select or clear **Display UV Isolines for wireframe display**.

    For models with curved faces, you may prefer to clear this selection to simplify the wire-frame display, so the rendering will be faster.

Rendering

1. To specify a default color for a Modeler drawing object or action (such as on select):
    - Select the object or action from the **Default color** pull-down list. Then click the color button to open the **Color** window.
    - Select a color, and click **OK**.
2. To specify how to render an object, select **WireFrame** or **SmoothShade** from the **Default view render** pull-down list.

    When dealing with complicated geometries, choose **WireFrame** rendering. This is faster than shaded rendering.
3. To use pre-defined material appearance if available, check the **Use material appearance if available** box. When this option is selected, the default color and transparency used for newly created objects come from material settings. It is possible that some materials, like custom materials, may not have a default appearance specified, in which case the default color and transparency specified under **Object Appearance** are used.
4. To set the **Default transparency**, move the slider, or enter a numerical value.
5. To set the **Default color**, click the color button to open the **Color** window.
6. To set the **Show Object Outline** , use the checkbox. Unchecking shows the objects but without the outline.
7. To set the **Object Visualization Outline** contrast, use the slider.The current value, on a range from 0 to 1 displays in the text field.

History Tree

1. Under **Default tree layout**:
    a. Select or clear the **Group objects by material** check box. This feature is the same as selecting or clearing the **Group Objects By Material** toggle command from the **Modeler** menu.
    b. Select or clear **Group components by definition**.
2. Under **Selection**, for the **Select last command on object select** option:
    - If checked, the history tree is expanded after operations on object properties, even if the tree is collapsed for the item.

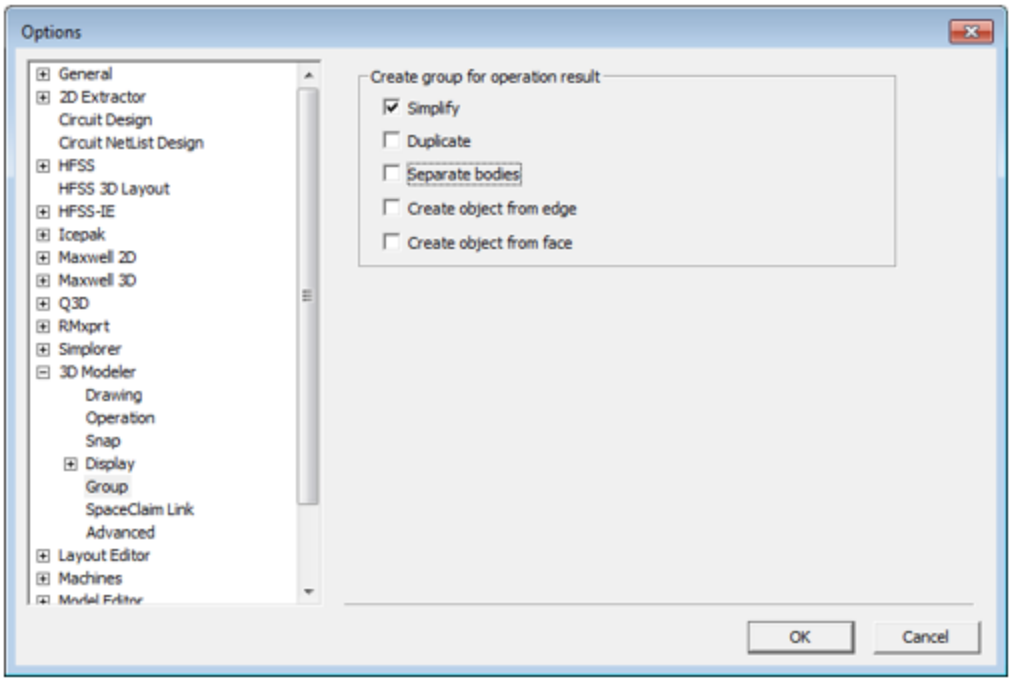
- If not checked, when you select an object in 3D view, only the object selected, and current tree collapse/expand state is preserved

3. Under **Selection**, for the **Expand history tree on object select** option:
   - If not checked, the history tree does not open on object selection. This can be useful for speeding the display of multiple object selections, or complex objects.
   - If checked, selecting an object automatically opens the history tree.

3D Modeler Options: Group Options

To set the Group creation for operation result settings for the 3D Modeler, use **Tools>Options>General Options** to open the **Options** dialog and select **3D Modeler**. Under **3D Modeler** options, select **Group**.

1. To specify Group creation for operation settings, select or clear the following check boxes in the **Snap Mode** section:
   - Simplify
   - Duplicate
   - Separate Bodies
   - Create object from edge
   - Create object from face



3D Modeler Options: SpaceClaim Link

These options are set on the **SpaceClaim Link** panel under **3D Modeler** in the **Options** dialog.

Import Options

- Select or clear **Import Solid Bodies**.
- Select or clear **Import Surface Bodies**.
- Select or clear **Import Parameters**, and enter the desired **Parameter Key**.
- Select or clear **Import Rendering Attributes**.
- Select or clear **Import Material Assignments**.
- Select or clear **Import suppressed for physics objects**. By default, this is unchecked and objects suppressed for physics are not imported.

3D Modeler Options: Advanced

These options are set on the **Advanced** panel under **3D Modeler** in the **Options** dialog.

Faceting

1. Select or clear **Incremental faceting (facet only modified faces of object)**.
2. Select or clear **Facet bodies by face using multiple processors**.

UDM/UDP Geometry Computation for Optimetrics Analysis

Select either **Engine computes the geometry** or **Desktop computes the geometry**.

Note	Geometry computation with CAD Integration (dynamic links) and geometry sharing in Ansys Workbench is always by Desktop.

Setting Fields Reporter Options

To set the **Fields Reporter** options:

1. Click **Tools>Options>General Options** to open the **Options** dialog box, displaying a hierarchical list of options. Click the + next to **Reporter** to expand the list and select **Fields Reported** to display the following list:
   - **Animation**
   - **Mesh Plot**
   - **Streamline Plot**
2. On the **Animation** panel:
   a. Specify whether to **Group Field Overlays by Type** (default, yes).
   b. Set the default Phase Animation settings for **Scalar Plots** and **Vector Plots.**

   Each of these accepts values for **From** and **To** in degrees, and the number of **Steps**.
3. On the **Mesh Plot** tab, in the **Clipping of volume mesh plot** section, choose the conditions under which the plot is updated dynamically while dragging the clip plane. Options are: **Never**, **Always**, and **When number of mesh element is less than** (default number is 50000).
4. On the **Streamline Plot** panel:

a. Set the **Streamline drawing stopping criteria.**
b. Set the **Streamline marker spacing** for the **Number of marker per bounding box diagonal**. The default value is 100.

Exporting Options Files

The options files at all levels that may affect a product running as a specific user on a specific host may easily be exported. You can export options files by selecting **Tools>Options>Export Options Files.** This brings up a browser dialog that you use to select the destination directory for the options files. Pressing the Cancel button will cancel the export command. Pressing the Open button will copy all of the config files for the current user and current host to the specified directory. Config files for the install, install_machine, user, and user_machine levels will be copied, if they exist. One additional file, admin.XML, will also be copied to the destination directory. This file does not contain user configurable options.

Related Topics

Setting Options in Maxwell
Setting Options via Configuration Files
Example Uses for Export Options Features
User Options and the Update Registry Tool
Batchoptions Command Line Examples

Setting Options via Configuration Files

> **Note:**
>
> Because Maxwell can interact with other Ansys products, the following sections on setting options via configuration files include examples and information applicable to various other Ansys products.

In addition to setting options from the Desktop UI, you can also set options in several configuration files.

Option settings in configuration files may apply to all users or only to a specific user, and may apply to all hosts or only to specific hosts. There are four levels, listed below from most specific (highest precedence) to most general (lowest precedence):

- Host-dependent user options (apply to the specified user on the specified host only)
- Host-independent user options (apply to the specified user on all hosts)
- Host-dependent default options (apply to all users on the specified host)
- Installation default (default for all users on all hosts)

A setting at any level will override settings at lower levels in the list above. If there is no setting in any file, the application default value will be used. See UpdateRegistry for instructions on selecting these levels.

> **Important:**
>
> Options set from the Desktop UI will override settings in configuration files.

Behavior Examples

Consider running an application as user **jsmith** on host **host123**. Also, assume the following conditions apply:

- There is no *host-dependent user setting* for the **Expand Project Tree on Insert** option in the *host-dependent user options* config file for user **jsmith** on host **host123**.
- But, the option exists in the *host-independent user options* config file for user **jsmith**.

In this situation, the latter setting is used (if it is not overridden using the Desktop UI). Any settings in the *host-dependent default options* config file or the *installation default* config file are ignored.

As another example, consider running an application as user **jdoe** on host **host123**. Also, assume that the following conditions apply:

- There is no setting for the **Expand Project Tree on Insert** option in the *host-dependent user options* config file for **jdoe** on **host123**.
- There is no setting in the *host-independent user options* config file for user **jdoe**.
- And, there is no setting in the *host-dependent default options* config file for host **host123**.

In this situation, the value from the *installation default* config file is used, if present.

Rules for Modifying Option Settings

Option settings displayed in the Desktop UI follow the above rules. That is, if there is a setting in any of the option config files, then the setting from the highest priority config file is displayed in the Desktop UI. If there is no setting in any of the option config files, then the global default value is used.

You can modify settings using the various **Options** dialog boxes accessed via the **Tools** > **Options** menu. If a dialog box is closed with the **Cancel** button, then changes made to any settings are discarded. If the dialog box is closed with the **OK** button, then any settings that have been changed become immediately effective. These changes are written to the host-dependent user options config file when you exit the Electronics Desktop application. The changed values written to this file are then used the next time that the application is launched by the same user on the same host. The Desktop UI option settings are not written to any of the other option config files.

Configuration File Locations

The configuration files for host-dependent and installation **default** options reside at: <*installation_directory*>\<*version*>\<*platform*>\config. The configuration files for host-dependent and host-

independent **user** options reside in a subfolder of the user's Documents folder (for Windows) or the user's HOME folder (for Linux). See the tables below for specific Windows and Linux file names and paths.

Products with Multiple Desktop Versions

For products that have multiple Desktop versions, each version has a separate user-specific **config** folder with a different value for <*ApplicationName&Version*> (the parent folder name). Simarly, each version has a separate **config** folder for the *host-dependent default* and *installation default* config files. This folder is under the path, <*InstallationDirectory*>\<*version*>\<*platform*>. See the tables below for specific Windows and Linux file names and paths.

Table of Directories and Files

The following table shows the directories and files, where the *Level Name* is the name used to describe an options config file when using the UpdateRegistry tool.

Config File	Level Name	File Name	Windows Directory Path	Linux Directory Path
host-dependent user options	user_machine	<*hostname*>_user.XML	%UserProfile%\Documents\Ansoft\<*ApplicationName&Version*>\config	$HOME/Ansoft/<*ApplicationName&Version*>/config
host-independent user options	user	user.XML		
host-dependent default options	install_machine	<*hostname*>.XML	<*InstallationDirectory*>\v<*version*>\Win64\config	<*InstallationDirectory*>/v<*version*>/Linux64/config
installation default	install	default.XML		

Note:

- ***<hostname>*** is the name of the computer on which the Electronics Desktop software is installed
- **%UserProfile%** is a Windows variable that represents the currently active user's profile (for example, C:\Users\JohnDoe)
- ***<ApplicationName&Version>*** is the product name (without spaces) followed by the four-digit year of the version, a decimal point, and the minor release number (such as ElectronicsDesktop2023.1)
- **$HOME** is the user's home directory on Linux
- ***<InstallationDirectory>*** is the root folder where the Electronics Desktop software is installed (typically, C:\Program Files\AnsysEM, on Windows, or /opt/AnsysEM, on Linux)
- ***<Version>*** is the last two digits of the product version's year followed by the minor release number, without a decimal point (such as 231)

The following table shows an example of specific file names and directory names for a typical Ansys Electronics Desktop installation on Microsoft Windows and on Linux. These are the files that apply to software version **2023 R1**, user "**jsmith**," and hostname "**host123**":

Config File	Level Name	File Name	Windows Directory Path	Linux Directory Path
host dependent user options	user_machine	host123_user.XML	C:\Users\jsmith\Documents\Ansoft\ElectronicsDesktop2023.1\config	/home/jsmith/Ansoft/ElectronicsDesktop2023.1/config
host independent user options	user	user.XML		
host dependent default options	install_machine	host123.XML	C:\Program Files\AnsysEM\v231\Win64\config	/opt/AnsysEM/v231/Linux64/config
installation default	install	default.XML		

Note:

As with the temporary file location configuration files, the settings in these options files have precedence in the following sequence: user_machine (highest precedence), user, install_machine, install (lowest precedence). The first time you start and then exit the application, the file at the "user_machine" level is created (<*hostname*>_user.XML). The other files are only created if you use the UpdateRegistry tool to specify an option at the "user," "install_machine," or "install" level. If the temporary directory is set to an empty string in a configuration file, then that setting is ignored.

Setting or Removing Option Values in Configuration Files: UpdateRegistry Command

UpdateRegistry is a command line tool used to modify option settings in the options config files. You can use this command to add, change or remove settings from any of the option config files. This tool is included in the installation directory of each product. This feature makes it easier for different users to use Ansys Electromagnetics tools installed on shared directories or network drives.

The UpdateRegistry command has multiple command line formats, as shown below.

The following command line options are *mutually exclusive*:

- `-Set`
- `-Get`
- `-GetKeys`
- `-Delete`
- `-FromFile`

UpdateRegistry -Set Command

This command is used to add or modify an option setting in an option config file. If the option config file does not exist, it will be created. If the setting does not exist in the specified config file, it will be added. If the setting already exists in the specified config file, then the value will be changed to the specified value.

Example:	`UpdateRegistry -Set -ProductName <name> -RegistryKey <keyPath> -RegistryValue <value> [ -RegistryLevel <level> ]`

Required:	<name>	The application or product name and version. For example, ElectronicsDesktop2023.1. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: ...\AnsysEM\<version>\Win64\config\
	<keyPath>	The pathname of the option setting. This includes the same analysis-related registry keys and values that are displayed by the -batchoptions help. For example, Desktop/Settings/ProjectOptions/AnimationMemory
	<value>	The new value of the option, typically a string or a number. If the value contains spaces, it must be quoted.
Optional:	<level>	When specifying -RegistryLevel, this is a string denoting which config file to modify. One of: install, install_machine, user, and user_machine. If the level is not specified, the user_machine (host-dependent user options) file is modified.

UpdateRegistry -Get Command

This command is used to view an option value in an option config file. If the setting exists in the specified config file or files, then the value, the value type and the config file where the value was found will be reported. If no value is found, then that will also be reported.

Example:	UpdateRegistry -Get -ProductName <name> -RegistryKey <keyPath> [-RegistryLevel <level>]
Required:	<name> The application or product name and version. For example, ElectronicsDesktop2023.1. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: \AnsysEM\<version>\Win64\config\
	<keyPath> The pathname of the option setting. This includes the same analysis-related registry keys and values that are displayed by the -batchoptions help. For example, Desktop/Settings/ProjectOptions/AnimationMemory.

Optional:	<level>	When specifying -RegistryLevel, this is a string denoting which config file to modify. One of: install, install_machine, user, and user_machine. If the level is not specified, then all config files are searched in order of precedence.

UpdateRegistry -GetKeys Command

This command is used to view the allowed key names for all of the option settings, or to view a subset of the key names that match a string. For each key displayed, the current value, if any, is also reported. If a key has a value in multiple config files, then only the highest precedence value is reported.

Example:	UpdateRegistry -GetKeys [<pattern>] -ProductName <name> [-Case]	
Required:	<name>	The application or product name and version. For example, ElectronicsDesktop2023.1. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: \AnsysEM\<version>\Win64\config\
Optional:	<pattern>	If no pattern is specified, then all allowed key names are reported. If a pattern is specified, then only keys that match the pattern are shown. For example, Settings/Project. If the name contains spaces, it must be quoted. By default, the pattern match is case insensitive.
	-Case	If this command line option is specified, then the pattern match is case sensitive.

UpdateRegistry -Delete Command

This command is used to remove an option setting from an option config file. If the setting does not exist in the specified config file, the file will not be changed. If the setting exists in the specified config file, then it will be removed. A setting may need to be removed from an option config file, to allow the setting from a lower priority file to be used by the application.

Example:	UpdateRegistry -Delete -ProductName <name> -RegistryKey <keyPath> [-RegistryLevel <level>]

Required:	`<name>`	The application or product name and version. For example, ElectronicsDesktop2023.1. If the name contains spaces, it must be quoted. The name can be found in the ProductList.txt file in the install directory: \AnsysEM\<version>\Win64\config\
	`<keyPath>`	The pathname of the option setting. This includes the same analysis-related registry keys and values that are displayed by the -batchoptions help. For example, Desktop/Settings/ProjectOptions/AnimationMemory.
Optional:	`<level>`	When specifying `-RegistryLevel`, this is a string denoting which config file to modify. One of: install, install_machine, user, and user_machine. If the level is not specified, the user_machine (host-dependent user options) file is modified.

UpdateRegistry -FromFile Command

You can use this form of the UpdateRegistry command to set multiple key-value pairs from a file with a single UpdateRegistry command. You specify the -FromFile command line option. This option must be followed by a filename. The file may contain multiple entries, where each entry contains a registry key and a registry value. The key-value pairs are added to the registry level specified by the -RegistryLevel command line option; if no -RegistryLevel is specified, then the default registry level (user_machine) is used.

UpdateRegistry File Format

> **Note:**
>
> Functionality featured in the example(s) in this section applies to multiple design types.

The file format is similar to the -batchoptions file format. An example UpdateRegistry file is shown below:

```
    $begin 'AddEntries'
'TempDirectory'='C:/temp/AnsysEM'
'Desktop/Settings/ProjectOptions/HPCLicenseType'='Pool'
    $end 'AddEntries'
```

Additional notes on the file format:

- The file may contain an arbitrary number of entries, one per line.
- Leading whitespace on each line is ignored. Spaces or tabs may be used to make the file more readable.

Registry key pathname:

- The registry key pathname appears before the equal sign "=" on each line.
- Each registry key pathname must be enclosed in single quotes.
- This includes the same analysis-related registry keys and values that are displayed by the -batchoptions help.

Registry value:

- The registry value appears after the equal sign on each line.
- Integral registry values must not be enclosed in quotes.
- All other registry values are treated as strings, and must be enclosed in single quotes.
- The forward slash "/" may be used as a directory separator on Windows and Linux. The back slash "\" may be used as a directory separator on Windows only.
- The back slash "\" is used as an escape character in the value string. That it, this character removes the special meaning of the following character.
- The single quote character normally ends the value string. The back slash may be used to remove this special meaning, and include a single quote in the string.
- To use a back slash as a directory separator on Windows, it must be escaped. That is, a double back slash "\\" is used to denote a single directory separator.

Alternative UpdateRegistry File Format:

- Analysis Configuration File format, which is exported from the HPC and Analysis Options dialog box.

> **Note:**
>
> If a current registry does not exist, Ansys Electronics Desktop can detect earlier minor versions of same application on the same machine. If such a registry exists (and does not involve -help, -batchoptionhelp, IsBatchMode(), -regserver, -unregserver, running a script, or non graphical mode), a prompt displays allowing you to port the registry from an earlier version.

Related Topics

Setting Options via Configuration Files
Example Uses for Export Options Features
User Options and the Update Registry Tool
Batchoptions Command Line Examples

Example Uses for Export Options Features

The **Tools** > **Options** > **Export Options** feature is intended to make it easier for different users to use Ansys Electromagnetics tools installed on shared directories or network drives. This section

outlines some use cases enabled by this feature.

> **Note:**
>
> Functionality featured in the examples in this topic apply to multiple design types.

Options that Apply to All Users

In many cases, an Ansys Electromagnetics tool installation is administered and maintained by a single user or group and used by a number of other users or groups. The permissions of the Ansys Electromagnetics tool installation may be set so that the administrator may add, delete or modify files, but other users may only read or execute these files. The administrator may set the recommended option settings in the installation default config file and/or the host dependent default options config file. These config files reside within the installation directory hierarchy, and should generally have the same permissions as other Ansys Electromagnetics tool installation files. This allows that administrator to control these settings, but does not allow other users to add, remove, or change these settings.

Each user can override any of these settings, if needed. This may be done using the Desktop UI, which affects the host-dependent user options config file. It may also be done using the host-independent user options config file. If user has overridden an option setting in either of the user files, the user may revert back to the option settings provided by the administrator by removing the setting of the same option in the host-dependent user option config file and/or the host-independent user option config file.

For global defaults, the administrator may set a value in the installation default config file. These settings will to apply to all users on all hosts.

In some cases, there are significant differences between the capabilities of different hosts. The host-dependent default config file may be used to specify different default values on some hosts. Any setting in a host dependent default config file would affect all users running on the specified host. The installation default value is used if there is no value specified for the setting in the host-dependent default config file for the current host. Note that the host-dependent default config file is named <*hostname*>.xml, where *hostname* is the name of the host computer.

Example: Searching for a Registry Key Pathname

Both administrators and ordinary users may occasionally use the UpdateRegistry command line tool to add, change or delete settings. To use this tool, the registry key pathname must be known by the user. The -GetKeys option may be used to quickly search for a key pathname if some information is known about it. For example, if the administrator knows that there is a setting related to issuing warning messages when available disk space is low, but does not know the exact key name, the following command may list some of the keys related to disk space:

```
UpdateRegistry -GetKeys disk -ProductName
ElectronicsDesktop2023.1
```

This will display a list of all keys that match the string "disk" case insensitively.

Typical output may look like the following:

```
Registry keys matching pattern <disk> case insensitively:
```

```
Desktop/Settings/ProjectOptions/DiskLimitForAbort: value is <0>
at level <user_machine>
```

Example: Setting an Installation Default Value

The normal default for the Options/General/Desktop Performance/Warn when available disk space is less than setting is 0 MB. If the administrator is concerned that running out of disk space might be a common problem, the administrator could set the installation default for the warn setting setting to 1000 MB, for example. This limit would then apply to all users running on all hosts. The administrator could use the following command to change this setting for Ansys Electronics Desktop:

```
UpdateRegistry -Set -ProductName ANSYSElectronicsDesktop2023.1
 -RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
 -RegistryValue 1000
 -RegistryLevel install
```

Example: Setting a Host-Dependent Default Value

For this example, assume that all hosts have two cores, except for three hosts: bighost1, bighost2, and bighost3, which have eight cores each. The administrator has set the Desktop/Settings/ProjectOptions/NumberOfProcessors option value to 2 in the installation default config file. The administrator may set the Desktop/Settings/ProjectOptions/NumberOfProcessors option value to 8 in the host-dependent default config files for the three hosts having 8 cores: bighost1, bighost2 and bighost3. The administrator may log in to host bighost1 and run the following command to change this setting for the host-dependent default options config file for host bighost1:

```
UpdateRegistry -set -ProductName ElectronicsDesktop2023.1
 -RegistryKey
Desktop/Settings/ProjectOptions/NumberOfProcessors
 -RegistryValue 8
 -RegistryLevel install_machine
```

To make this change for the other two hosts, the administrator would log in to bighost2 and bighost3, and run the same command on each of those hosts.

Example: Reverting from a User-Defined Option Value to the Administrator Default

Consider the case in which Electronics Desktop was installed and the administrator initially did not set a value for the Desktop/Settings/ProjectOptions/DiskLimitForAbort setting in the default installation config file. User jsmith (who always uses host jshost) wanted to be warned before disk space dropped to zero, so he set the Desktop/Settings/ProjectOptions/DiskLimitForAbort to 100 MB using the UI. This setting is recorded in the host dependent user options config file for host jshost and user jsmith. Now the administrator learns that many users are running into disk space issues, so that administrator sets the installation default value for the setting Desktop/Settings/ProjectOptions/DiskLimitForAbort to 1000 MB, as in the above example.

When user jsmith runs Electronics Desktop on host jshost, the disk limit is 100 MB, not 1000 MB, because the host-dependent user options config file overrides all of the other config files. User jsmith may revert to the administrator provided default by removing this setting from the host

dependent user options config file for host jshost and user jsmith. The following command may be run by user jsmith on host jshost to remove this setting:

```
UpdateRegistry -Delete -ProductName ElectronicsDesktop2023.1
 -RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
 -RegistryLevel user_machine
```

If user jsmith had added a value for this setting to the host-independent user options config file, then user jsmith would also run the following command to remove this setting from the host-independent user options config file:

```
UpdateRegistry -Delete -ProductName ElectronicsDesktop2023.1
 -RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
 -RegistryLevel user
```

User Options and the Update Registry Tool

When you change an options value using the Desktop UI, the new value is stored in the host-dependent user options config file. You can also use the UpdateRegistry tool to add or modify settings in the host-dependent user options config file. You cannot use the Desktop UI to remove settings from the host-dependent user options config file, however. You must use the UpdateRegistry tool to remove settings from the host-dependent user options config file.

If a user has not explicitly created a host-dependent user options config file or a host-independent user options config file, then when a user first runs an Ansys Electromagnetics tool on a host, all settings will come from the host-dependent default options config file or the installation default options config file. Any settings for another host in a host-dependent user options config file will not be carried over to the new host. This may be inconvenient if the user has preferred option settings that differ from the settings that apply to all users, especially if the user runs the Ansys Electromagnetics tool on a number of different hosts. In this case, the user may set these option values in the user's host-independent user options config file. Then, these option values will be used on all new hosts, overriding any values set by the administrator to apply to all users. Any changes made in the UI will only affect the user's host-dependent user options config file for the current host.

Getting a Value from a Specific Configuration File

Getting a Value Using Precedence Rules

Example of Removing a Host Dependent User Option Setting

Example Adding a Host Independent User Option Setting

Setting the Temporary Directory

Temporary Directory Configuration File Format

Example Temporary Directory Configuration File Format

Setting the Temporary Directory Using the GUI

Setting or Removing Temporary Directory Values in Configuration Files: UpdateRegistry Command

Getting a Value from a Specific Configuration File

In the previous example, the user jsmith may decide to check the Desktop/Settings/ProjectOptions/DiskLimitForAbort setting in the host independent user configuration file before making any changes to this setting. The following command may be used to quickly view this setting before making the change:

```
UpdateRegistry -Get -ProductName ElectronicsDesktop2023.1 -
RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort -
RegistryLevel user
```

Getting a Value Using Precedence Rules

In may cases, the user is more interested in the value of a setting that will be applicable when running the product than in the setting in a single configuration file. If the -Get option is used with no -RegistryLevel specified, then the value reported is the value found in the highest precedence configuration file. If the user jsmith is interested in the highest precedence value for the Desktop/Settings/ProjectOptions/DiskLimitForAbort setting, then the following command may be used to report this information:

```
UpdateRegistry -Get -ProductName ElectronicsDesktop2023.1 -
RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
```

Example of Removing a Host Dependent User Option Setting

For this example, user **jsmith** always uses host **jshost** to run Electronics Desktop. At some point, **jsmith** set the Autosave interval in the **General Options** dialog box, **Project Options** tab to 1000 edits, and this value was written to the jsmith's host dependent user options config file for host jshost. Now, **jsmith** wants to remove this setting and return to the default value of 10. User **jsmith** may run the following command on host jshost to remove the **Desktop/Settings/ProjectOptions/AutoSaveInterval** option value from this config file:

```
UpdateRegistry -Delete -ProductName ElectronicsDesktop2023.1
-RegistryKey Desktop/Settings/ProjectOptions/AutoSaveInterval
-RegistryLevel user_machine
```

Related Topics

User Options and the Update Registry Tool

Example Adding a Host Independent User Option Setting

Setting the Temporary Directory

Example Temporary Directory Configuration File Format

Setting the Temporary Directory Using the GUI

Example Adding a Host Independent User Option Setting

Consider the case in which there is no value set for the **Desktop/Settings/ProjectOptions/DiskLimitForAbort** setting for all users. The default is then 0 MB. User **jsmith** uses a variety of hosts and wants to be warned whenever disk space drops to 250 MB on any host. User **jsmith** may use the following command to set the **Desktop/Settings/ProjectOptions/DiskLimitForAbort** option value to 250 MB for all hosts:

```
UpdateRegistry -set -ProductName ElectronicsDesktop2023.1
-RegistryKey Desktop/Settings/ProjectOptions/DiskLimitForAbort
-RegistryValue 250 -RegistryLevel user
```

Related Topics

User Options and the Update Registry Tool

Setting the Temporary Directory

Temporary Directory Configuration File Format

Setting the Temporary Directory Using the GUI

Setting the Temporary Directory

The temporary directory may be viewed or set using the Electronics Desktop user interface (UI), or from the command line.

To set the directory via the UI:

Navigate to **Tools** > **Options** > **General Options**. In the tree at the left side of the dialog box, expand the **General** branch and select **Directories**. Then, use the **Temp** field and **Override** check box to enter a desired directory path. Values set in this manner are written to the user_machine level configuration file for the temporary directory. If the **Override** check box is cleared, clicking **OK** changes the user_machine level setting for the temporary directory to an empty string. This enables settings from the next highest precedence configuration file. The file that provides the currently active temporary directory setting is shown under the **Temp** edit box.

To set the temporary directory from the command line, using the -batchoptions command line option. See: Running Ansys Electronics Desktop from a Command Line and -Batchoptions Command Line Examples.

The temporary directory may be configured with an installation default value, as well as a host-dependent default value, a host-independent user-specified value and a host-dependent user-specified value. Temporary directory settings are stored in different files from the other option settings. These files are located in the same directories as the configuration files for the other option settings. The following table shows the directories and files used to store temporary directory settings.

Config File	Level Name	File Name	Windows Directory Path	Linux Directory Path
Host-dependent, user-specific temporary directory	user_machine	<*hostname*>.cfg	%UserProfile%\Documents\Ansoft\<*ApplicationName&Version*>\config	$HOME/Ansoft/<*ApplicationName&Version*>/config
Host-independent, user-specific temporary directory	user	default.cfg		
Host-dependent, default temporary directory	install_machine	<*hostname*>.cfg	<*InstallationDirectory*>\v<*version*>\Win64\config	<*InstallationDirectory*>/v<*version*>/Linux64/config
Installation default temporary directory	install	default.cfg		

Note:

- **<*hostname*>** is the name of the computer on which the Electronics Desktop software is installed
- **$HOME** is the user's home directory on Linux
- **<*ApplicationName&Version*>** is the product name (without spaces) followed by the four-digit year of the version, a decimal point, and the minor release number (such as ElectronicsDesktop2023.1)
- **%UserProfile%** is a Windows variable that represents the currently active user's profile (for example, C:\Users\JohnDoe)
- **<*InstallationDirectory*>** is the root folder where the Electronics Desktop software is installed (typically, C:\Program Files\AnsysEM, on Windows, or /opt/AnsysEM, on Linux)
- **<*Version*>** is the last two digits of the product version's year followed by the minor release number, without a decimal point (such as 221)

As with other options, the settings in these files have precedence in the following sequence: user_machine (highest precedence), user, install_machine, install (lowest precedence). The installer creates the file at the "install" level (default.cfg). The first time you start and then exit the application, the file at the "user_machine" level is created (<*hostname*>.cfg). The other files are only created if you use the UpdateRegistry tool to specify an option at the "user" or "install_machine" level. If the temporary directory is set to an empty string in a configuration file, then that setting is ignored.

Related Topics

User Options and the Update Registry Tool
Example Adding a Host Independent User Option Setting
Example Temporary Directory Configuration File Format
Setting the Temporary Directory Using the GUI
Temporary Directory Configuration File Format

Temporary Directory Configuration File Format

This section describes the format of the Temporary Directory configuration files. The format is the same for files at all four levels: user_machine, user, install_machine, and install. These files are text files, so any text editor may be used to modify or create Temporary Directory configuration files.

An example temporary directory configuration file is shown below:

```
$begin 'Config'
```

```
tempdirectory='C:/TEMP/AnsysEM'
$end 'Config'
```

The temporary directory specified by this configuration file is C:/TEMP/AnsysEM.

Additional notes:

The string containing the pathname of the temporary directory must be enclosed in single quotes.

The forward slash "/" may be used as a directory separator on Windows and Linux. The back slash "\" may be used as a directory separator on Windows only.

The back slash "\" is used as an escape character in the tempdirectory string. That it, this character removes the special meaning of the following character.

The single quote character normally ends the tempdirectory string. The back slash may be used to remove this special meaning, and include a single quote in the string.

To use a back slash as a directory separator on Windows, it must be escaped. That is, a double back slash "\\" is used to denote a single directory separator.

On Windows, a UNC path normally begins with two back slash characters. In a tempdirectory string, each of these back slash characters must be doubled, so four consecutive back slashes "\\\\" are used in the config file.

UNC Example

Config file:

```
$begin 'Config'
tempdirectory='\\\\hostxyz\\TEMP\\abc'
$end 'Config'
```

Here hostxyz is a host with a sharename TEMP having subdirectory abc used as the Temporary Directory. This shows that four back slashes are required for UNC names and that back slashes used as directory separators must be doubled.

Single Quote Example

Config file:

```
$begin 'Config'
tempdirectory='C:/TEMP/ab\'cd'
$end 'Config'
```

Temporary directory is C:/TEMP/ab'cd. This shows how to include a single quote in a tempdirectory pathname. It also shows that forward slashes may be used as directory separators on Windows.

Related Topics

User Options and the Update Registry Tool

Example Adding a Host Independent User Option Setting

Example for Setting the Temporary Directory

Setting the Temporary Directory Using the GUI

As for other options, the Temporary Directory may be viewed or set using the Desktop GUI. In the **Tools>Options>General Options** dialog box, the Temp Directory setting appears in the Directories group box on the **Project Options** tab. Activating the **Override** check box allows you to enter a desired directory pathname in the edit box or to click on the [...] button to bring up a directory file browser dialog box, from which you can select a temp directory. Values set in this manner are written to the user_machine level configuration file for the Temporary Directory. If the **Override** check box is unchecked, then when the **OK** button is pressed, the **user_machine** level setting for the Temporary Directory is changed to an empty string. This enables setting from the next highest precedence config file. The config file which provides the currently active Temporary Directory setting is shown under the Temp Directory edit box in the **Project Options** tab of the **General Options** dialog box, if the **Override** check box is unchecked.

Setting the Temporary Directory From the Command Line

The temporary directory may be set from the command line, using the **-batchoptions** command line option. See Running Maxwell from a Command Line. The Batchoptions Command Line Examples section below includes examples that show how to set the Temporary Directory from the command line.

Related Topics

User Options and the Update Registry Tool
Example Adding a Host Independent User Option Setting
Example for Setting the Temporary Directory
Temporary Directory Configuration File Format
Example Temporary Directory Configuration File Format
Batchoptions Command Line Examples
Running Maxwell from a Command Line

Setting or Removing Temporary Directory Values in Configuration Files: UpdateRegistry Command

The UpdateRegistry command line tool, described above, may be used to view, add, change or remove the Temporary Directory setting from any of the Temporary Directory config files. The registry key for viewing or modifying the Temporary Directory is TempDirectory. The -Get, -Set, and -Delete options are valid for viewing a Temporary Directory setting, adding or changing a

Temporary Directory setting, or deleting a Temporary Directory setting. The -GetKeys option does not list the Temporary Directory key.

Related Topics

User Options and the Update Registry Tool

Batchoptions Command Line Examples

The **-batchoptions** entries command line argument may be used to specify one or more batchoptions settings on the command line. To specify multiple entries using a single **-batchoptions** argument, the entries should be enclosed in double quotes. Alternatively, the batchoptions may be specified in a file using the **-batchoptions** <filename> command line argument format. In this case, the filename is an absolute or relative pathname of the file containing the batchoptions, as described above. The two approaches may not be combined: either all batchoptions must be in a file or all batchoptions must be specified explicitly on the command line.

Batchoptions File Format

Example -BatchOptions with -Remote (Windows)

Example -Batchsolve with -Machinelist (Windows)

Example -Batchsolve with -Machinelist (Linux)

Example -Batchsolve for Local (Windows)

Related Topics

Running Maxwell From a Command Line

Batchoptions File Format

> **Note:**
>
> Functionality featured in the example in this section applies to multiple design types.

An example batchoptions file is shown below:

```
$begin 'Config'
  'Desktop/ProjectDirectory'='C:/test/projects'
  'Desktop/Settings/ProjectOptions/NumberofProcessors'=2
$end 'Config'
```

Additional notes on the file format:

- The file may contain an arbitrary number of batchoption entries, one per line.
- Leading whitespace on each line is ignored. Spaces or tabs may be used to make the file more readable.
- The **Registry Key** appears before the equals sign (=) on each line and must be enclosed in single quotes ('). The registry key includes the key path and the name of the registry value.
- The registry value (or option value) appears after the equals sign on each line.
- Registry keys are case-insensitive.
- There are two supported types of registry values—string and integer:
  - Each **string** value must be enclosed in single quotes (').
  - Do **not** enclose **interger** values in quotes.
- For file paths within string values, the forward slash (/) may be used as a directory separator on both Windows and Linux systems.
- Alternatively, on Windows only, the customary backslash (\ ) may be used as a directory separator. However, in string values, the backslash is used as an escape code for indicating special characters that cannot be typed directly (such as \n for a new line). Therfore, if you use the backslash as a directory path separator, each instance must be doubled (\\ ). An example is: '\\\\host3\\temp\\Ansoft'. In this case, each double backslash is interpreted as a single backslash (\\host3\temp\Ansoft).
- The single quote character (') normally ends a string value. If you need to include a single quote (or apostrophe) within a string, use the backslash-apostrophe (**\'**) escape sequence. For example, the string '%UserProfile%/Documents/Ansoft/John\'s_Files', is interpreted as: %UserProfile%/Documents/Ansoft/John's_Files.

Example -BatchOptions with -Remote (Windows)

> **Note:**
>
> Functionality featured in the examples in this section applies to multiple design types.

In this example, we run a batch Maxwell analysis of project file project1.aedt which contains a 3D design. We want all temporary files and directories created in directory C:\temp\Maxwell instead of using the installation default for the Temporary Directory. We decide that the analysis will be done on a remote host, at IP address 12.34.56.78. Because of limited memory on the remote host, we decide to run the analysis using only a single COM engine. Because the remote host has four cores, we decide to use four processors for the analysis. We can use the **-Remote** option to specify that there will be a single remote COM engine.

Here is a sample command line for this analysis, where the project file \\somehost\projects\project1.aedt is located in a shared directory specified using a UNC path:

```
ansysedt -BatchSolve -Remote -Machinelist list=12.34.56.78
 -batchoptions "'TempDirectory'='C:/temp/Maxwell'
 'Desktop/Settings/ProjectOptions/NumberofProcessors'=4"
 \\somehost\projects\project1.aedt
```

An alternative is to use the **-Distributed** command line option. Because the -Machinelist list contains only one host, there is a single remote COM engine in this case also.

```
ansysedt -BatchSolve -Distributed -Machinelist list=12.34.56.78
 -batchoptions "'TempDirectory'='C:\\temp\\Maxwell'
 'Desktop/Settings/ProjectOptions/NumberofProcessors'=4"
 \\somehost\projects\project1.aedt
```

The second line of the first example shows that you can use the forward slash "/" as a Windows directory separator within option value 'strings'. In this case, it is used in the TempDirectory path. You can also use the customary backslash "\" as a Windows directory separator, but it must be doubled to "\\" because the backslash is also an escape character within parameter strings. This usage is demonstrated in the second line of the second example, again in the TempDirectory path.

Related Topics

Batchoptions Command Line Examples

Example -Batchsolve with -Machinelist (Windows)

Example -Batchsolve with -Machinelist (Linux)

Example -Batchsolve for Local (Windows)

Example -Batchsolve with -Machinelist (Windows)

Suppose that we want to run a batch Maxwell analysis of project file project1.aedt. Because all of our hosts have multiples of 2 cores, we specify that we will use two threads for multiprocessing for both the distributed (NumberOfProcessorsDistributed) and non-distributed (NumberOfProcessors) parts of the job. The analysis contains a sweep that will be distributed across three hosts, adam, bill, and charlie. The hosts adam and bill have four cores each, so we run two distributed COM engines on each of these hosts, each using two threads. Host charlie has only two cores, so we specify only one distributed COM engine on this host. This COM engine will also use two threads. We specify a desired RAM limit of 6 GB and a maximum RAM limit of 8 GB for this analysis. The RAM limits are specified in KB, so the desired RAM limit is 6291456 KB, and the maximum RAM limit is 8388608 KB.

Here is a sample command line for this analysis, where the project file \\dennis\projects\project1.aedtis located in a shared directory specified using a UNC path:

```
maxwell -BatchSolve -Distributed
-Machinelist list=adam,adam,bill,bill,charlie
-batchoptions "Maxwell/Preferences/MemLimitHard=8388608
Maxwell/Preferences/MemLimitSoft=6291456
Maxwell/Preferences/NumberOfProcessors=2
Desktop/Settings/ProjectOptions/NumberOfProcessors=2"
\\dennis\projects\project1.aedt
```

Related Topics

Batchoptions Command Line Examples

Example -BatchOptions with -Remote (Windows)

Example -Batchsolve with -Machinelist (Linux)

Example -Batchsolve for Local (Windows)

Example -Batchsolve with -Machinelist (Linux)

In this example, we run a batch Maxwell analysis of project file project2.aedt, which contains an Maxwell design. We have four identical hosts **host1**, **host2**, **host3**, and **host4** for analysis, and each host has 4 cores. We do not use multiprocessing for the distributed analysis, so **NumberOfProcessorsDistributed=1**. As each host has four cores, we specify multiprocessing using 4 threads for the non-distributed part of the analysis, so **NumberOfProcessors=4**. Because we do not use multiprocessing for the distributed analysis, we will run four distributed COM engines on each host, with a singe core available for each engine. As in Example 1, we specify a desired RAM limit of 6 GB and a maximum RAM limit of 8 GB for this analysis. The RAM limits are specified in KB, so the desired RAM limit is 6291456 KB, and the maximum RAM limit is 8388608 KB.

Here is a sample command line for this analysis, where the project file **/home/jsmith/projects/project2.aedt** is located in a shared directory:

```
maxwell -BatchSolve -Distributed
-Machinelist file=/home/jsmith/hosts/list2
-batchoptions "Maxwell/Preferences/MemLimitHard=8388608
Maxwell/Preferences/MemLimitSoft=6291456
Maxwell/Preferences/NumberOfProcessors=4
Desktop/Settings/ProjectOptions/NumberOfProcessors=1"
/home/jsmith/projects/project2.aedt
```

For this example, the hostnames are in the text file **/home/jsmith/hosts/list2**. Here are the file contents:

```
host1
host1
host1
host1
host2
host2
host2
host2
host3
host3
```

```
host3
host3
host4
host4
host4
host4
```

Related Topics

Batchoptions Command Line Examples

Example -BatchOptions with -Remote (Windows)

Example -Batchsolve with -Machinelist (Windows)

Example -Batchsolve for Local (Windows)

Example -Batchsolve for Local (Windows)

In this example, we run a batch Maxwell analysis of project file testproject.adsn on the local host. We want all temporary files and directories created in directory **C:\temp\maxwell** instead of using the installation default for the Temporary Directory. Because the local host has four cores, we decide to use four threads for multiprocessing, for both distributed and non-distributed parts of the analysis.

Here is a sample command line for this analysis, where the project file **\\host123\projects\testproject.aedt** is located in a shared directory specified using a UNC path:

```
maxwell -BatchSolve -Local -batchoptions
"TempDirectory='C:/temp/maxwell'
Maxwell/SolverOptions/NumProcessors=4
Maxwell/SolverOptions/NumProcessorsDistrib=4"
\\host123\projects\testproject.aedt
```

Related Topics

Batchoptions Command Line Examples

Example -BatchOptions with -Remote (Windows)

Example -Batchsolve with -Machinelist (Windows)

Example -Batchsolve with -Machinelist (Linux)

Batchoptions and Analysis Configurations in the Registry

Analysis configurations are used to specify machines and options for local, remote, and distributed analysis, including capabilities that are enabled by HPC licenses.

How Analysis Configurations are Stored in the Registry

A configuration contains information beyond the machine or machines to use for a solution. Examples are the number of processors to allocate to the analysis for each machine in the list, memory limits, directory locations for personal libraries and temporary files, and many other preferences.

Note:

- Options are arranged as keys and values (in a structure similar to the *Windows Registry*). However, these options are not a part of the *Windows Registry* but are separately stored and maintained by the Ansys Electronics Desktop software. For more information concerning the configuration files comprisisng the options registry, see the following topics:

  Setting Options Via Configuration Files
  Setting the Temporary Directory
- For access to options and functionality beyond what is directly accessible via the user interface or batch options, refer to the documentation of the **UpdateRegistry** tool. This tool is discussed in the following help topic and in the topics that follow it in the same branch of the product help:

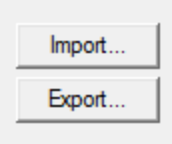
  Setting or Removing Option Values in Configuration Files: UpdateRegistry Command

General settings are associated with the Desktop application or 3D Editor. "HPCLicenseType" and "tempdirectory" values are at the root level of the registry. Other options are specific to a particular product or design type. For example, certain mesh, boundary, and memory limit settings as well as many other preferences are product-specific and are therefore associated only with the applicable design types. Such options appear with a consistent value name but in a different registry path for each applicable design type.

Copying a Configuration from one Design Type or Product to Another

To copy a configuration from one design type (or product) to another:

1. Click **Tools** > **Options** > **HPC and Analysis Options**. The *HPC and Analysis Options* dialog box appears.
2. On the **Configurations** tab of the **HPC and Analysis Options** dialog box, use the **Export…** button to export the configuration to a file.



3. Switch to the destination design type (or product) and use the **Import…** button to import the configuration data.

Any data that is not applicable to the destination design type is ignored; any settings present in the destination design type that were not present in the source configuration will be assigned default values. The user may then edit the copy, as desired.

Using HPC and Analysis Options for Configurations

Due to the complexity of the registry values for the configurations we do not recommend directly editing these values using the *UpdateRegistry* tool. Instead, use the **HPC and Analysis Options** dialog box to edit or create a configuration. (See: Setting HPC and Analysis Options.) Configurations created or edited using the GUI are stored in the **user_machine** level of the registry. You can create a configuration for one of the other registry levels the following steps:

1. Create the configuration using the Analysis Configurations GUI, then export the configuration to a file.
2. Delete the configuration using the GUI so that it will not be present in the **user_machine** level. Then, exit the GUI.
3. Use the **UpdateRegistry** tool to import the data into the desired registry level using the **-FromFile** option to specify the file exported via the GUI, and using the **-RegistryLevel** option to specify the registry level where the configuration is to be stored. For example, an administrator may use this approach to create a configuration at the **install** level that may be used by any user on any machine.

Batch Options

There is a large number of both general and product-specific options supported by the software. These options have evolved over time. Therefore, older batchoptions files may no longer be valid, and the options listed in the user interface of the current software may differ from earlier versions you have used. Additionally, there are options beyond those listed within the user interface (that is, the UI provides a subset of the available options). You can generate a more complete list of options by running the UpdateRegistry tool with the -GetKeys switch and piping the output to a text file, as detailed in the following Windows procedure:

1. In a command window, navigate to the following folder:

   <*installation_directory***>\v<***version***>\Win64**

2. Type and enter the following command:

   UpdateRegistry -GetKeys -ProductName ElectronicsDesktop20*xx.y* **>** <*text_file_path*>**\Batchoptions.txt**

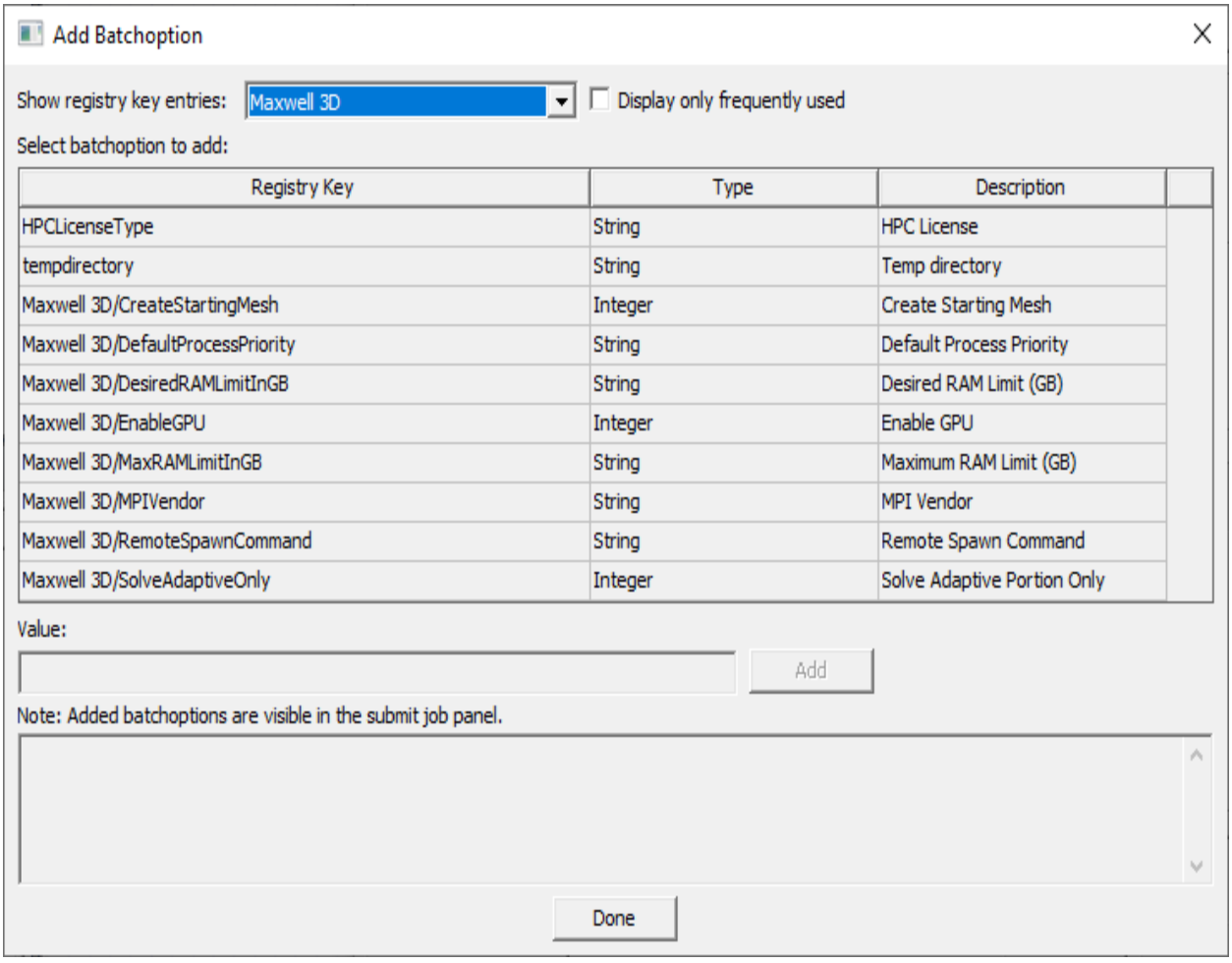
   Substitute the last two digits of the installed product version year and the minor release number for *xx.y* (such as **22.1**). Also, substitute the desired *text_file_path* (such as **%UserProfile%\Desktop**).

   The resulting **Batchoptions.txt** file will contain a nearly complete list of available options.

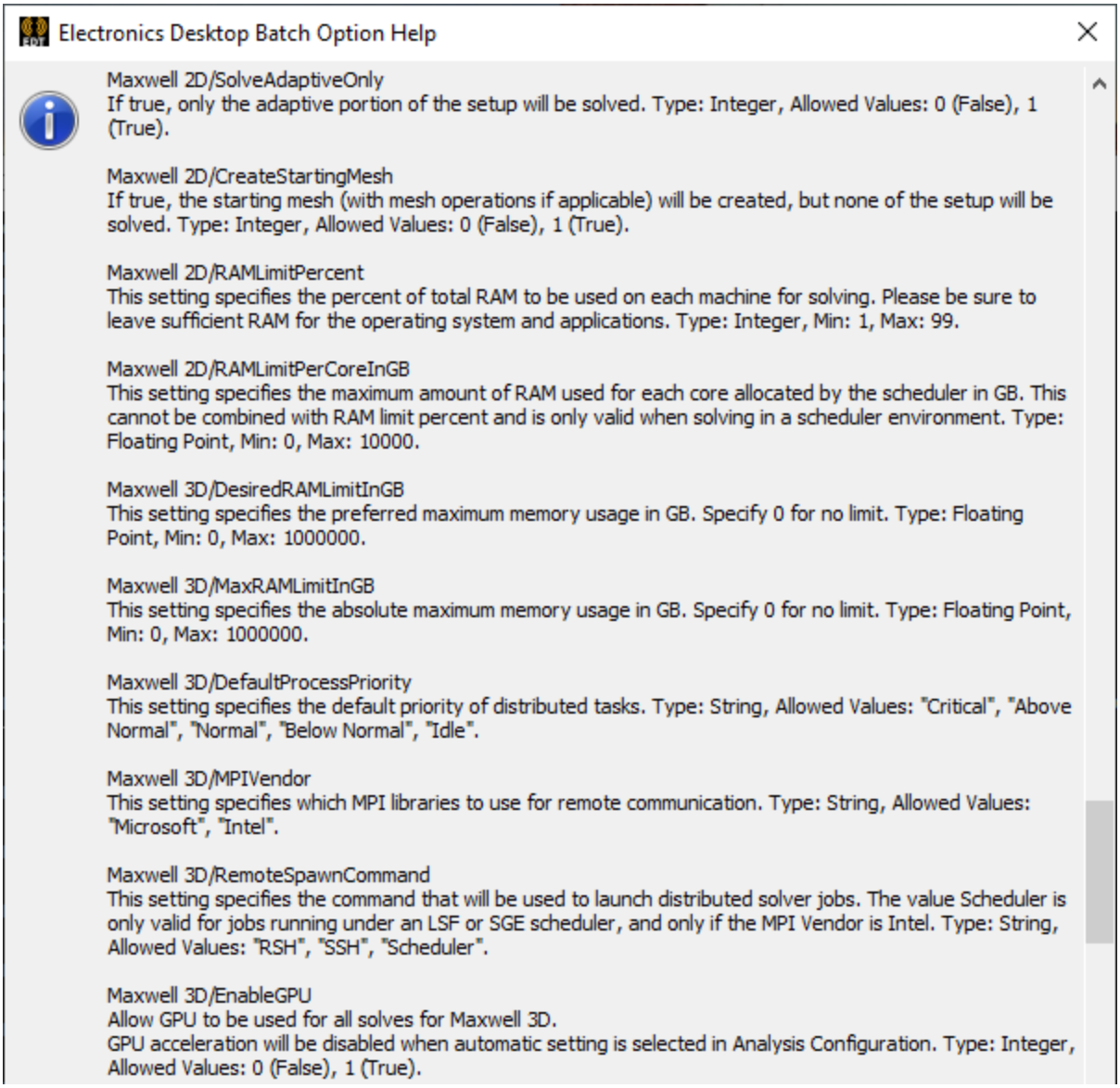
This procedure also works from the <*installation_directory*>/v<*version*>/Linux64 folder on the Linux platform.

> **Note:**
>
> For special product-specific options that are neither available from the GUI nor listed by the UpdateRegistry tool, see Special Batch Options. Even though the -GetKeys switch does not list these special options, you can still use the UpdateRegistry tool to set them.

When you submit jobs to a remote computer or cluster, you can specify batch options using the job submission GUI. When using the GUI, you select the batch options from a list and therefore avoid typographical errors. For the most commonly used batch options, there is detailed information about the allowed values. Click **Submit** on the **Simulation** ribbon tab to access the *Submit Job To* dialog box. Then, click **Add** to access the **Add Batchoption** dialog box pictured below:



To assist users who need to specify batch options and are unable to use the job submission GUI, a new help option has been added. If the Electronics Desktop application is launched with the **-batchoptionhelp** command line argument, a message box is displayed which lists and describes the most common design type-specific batch options:



Setting Analysis Configurations Using the User Interface

The Desktop User Interface may be used to select the configuration to be used for each design type. Each configuration is identified by a unique name. These settings may be viewed and modified using the **HPC and Analysis Options** dialog and modified using the **Analysis Configuration** dialog box.

See Setting HPC and Analysis Options.

Working with PEmag Designs

If **PEmag** is installed on your system, you can launch the PEmag desktop directly from the main Maxwell **Tools** menu. After you have created or opened a design in PEmag, PEmag can be used to generate both Maxwell 2D Electrostatic designs, and Eddy current designs. Also, provided that both Maxwell and PEmag desktops remain open, you can modify the related Maxwell design using PEmag.

The first time you invoke PEmag from Maxwell, an “About PExprt Modeler [PEmag]” dialog displays, informing you that modeling of magnetic components is disabled when PEmag is invoked from Maxwell. Uncheck the “I want to see this message...” checkbox to prevent this dialog from displaying again.

Note	Please refer to the documentation included with PEmag for detailed information on how to use that application to work with PEmag designs.

Related Topics

Creating Maxwell Designs Using PEmag

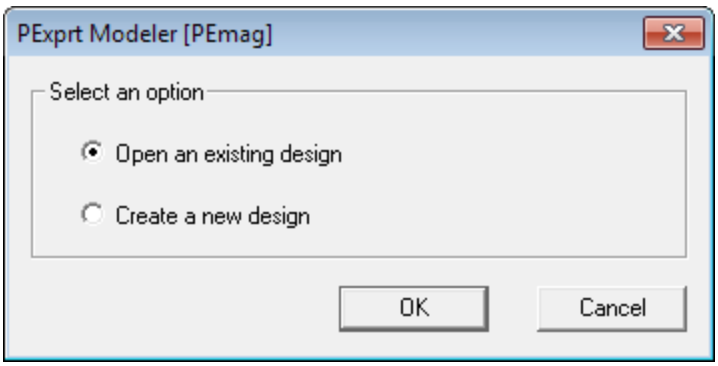
Modifying Maxwell Designs Using PEmag

Solving Maxwell Designs Generated by PEmag

Creating Maxwell Designs Using PEmag

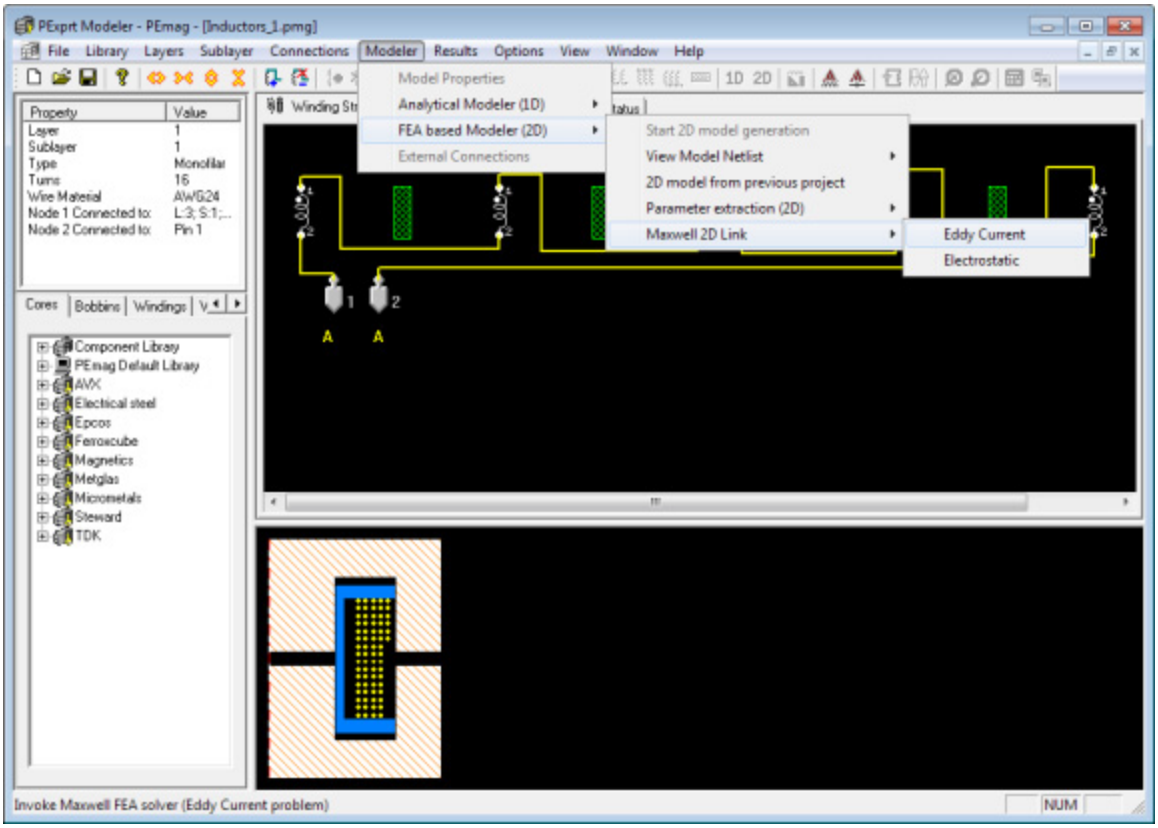
To create a Maxwell Design using PEmag:

1. Select **Tools>PEmag** to launch the PEmag desktop application.
2. In PEmag, choose either to **Open an existing design**, or to **Create a new design**.

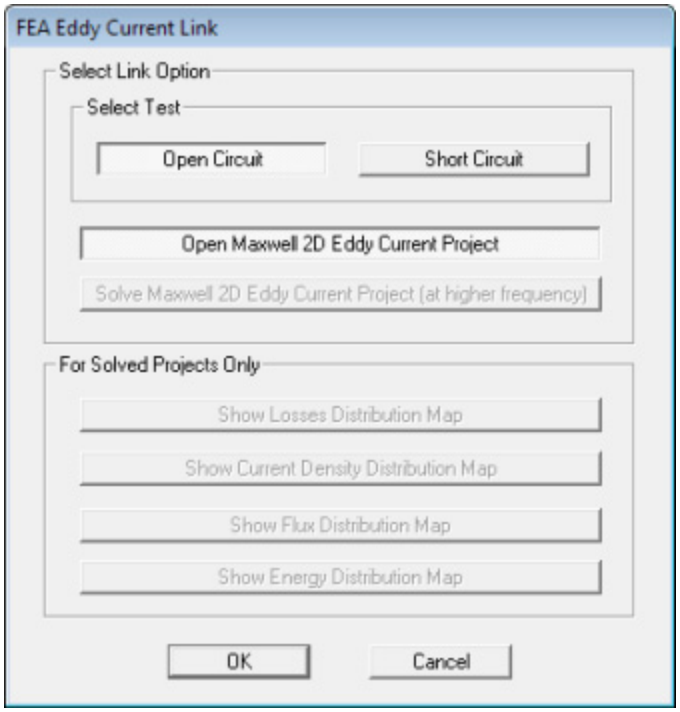


3. When finished creating the PEmag design, you can choose either **Modeler>FEA based Modeler(2D)>Maxwell 2D Link>Eddy Current** or **Modeler>FEA based Modeler (2D)>Maxwell 2D Link>Electrostatic** to have PEmag generate a corresponding design for Maxwell.

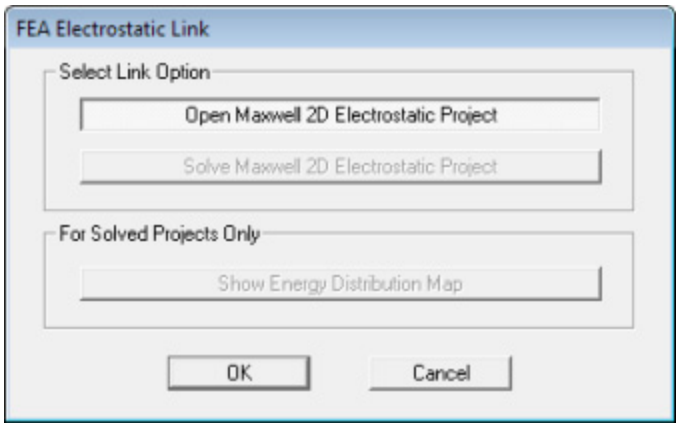
   You can also click on the **FEA Eddy Current** or **FEA Electrostatic** toolbar icons.



a. If you choose **Modeler>FEA based Modeler(2D)>Maxwell 2D Link>Eddy Current** a dialog similar to the following displays.



b. If you choose **Modeler>FEA based Modeler(2D)>Maxwell 2D Link>Electrostatic** a dialog similar to the following displays.



4. Click **OK** to close the dialog box. PEmag generates a Maxwell 2D design and adds it to the Maxwell Project Manager and desktop. The new design is created corresponding to the PEmag model. The project is immediately saved on disk with the name, *<PEmag project name>*.aedt, in a new folder named *<PEmag project name>*.pjt, which is under the same directory as the PEmag project.

Note	• Adding a Maxwell 2D design via PEmag is not undo-able. • After a Maxwell design is created through PEmag, attempting to close Maxwell while PEmag is still open, displays a message to inform that the

	Maxwell desktop cannot be closed while PEmag is still open.

Related Topics

Modifying Maxwell Designs Using PEmag

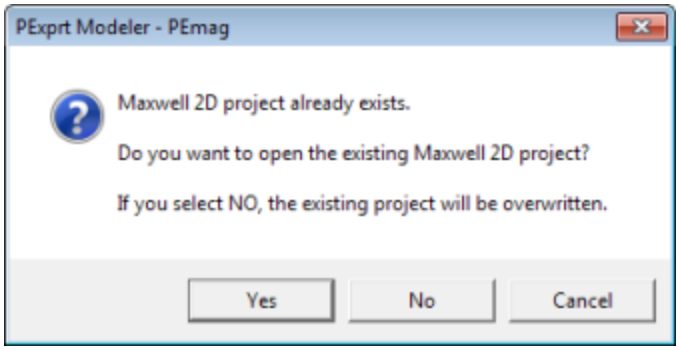
Solving Maxwell Designs Generated by PEmag

Modifying Maxwell Designs Using PEmag

After a Maxwell project is created through PEmag, the Maxwell project and design can be modified in the Maxwell desktop while the PEmag desktop is still open. The PEmag model can also be modified through PEmag desktop while the Maxwell desktop is open.

Note	• Modifying a Maxwell 2D design via PEmag is not undo-able. • After a Maxwell design is modified through PEmag, attempting to close Maxwell while PEmag is still open, displays a message to inform that the Maxwell desktop cannot be closed while PEmag is still open. • Modifications made in the Maxwell project are not updated in the PEmag project.

- If you modify the Maxwell 2D design, or add a new Maxwell design to the project in Maxwell; or if you modify the PEmag model in the PEmag desktop and regenerate the Maxwell design (by selecting either **Modeler>FEA based Modeler(2D)>Maxwell 2D Link>Eddy Current** or **Modeler>FEA based Modeler(2D)>Maxwell 2D Link>Electrostatic**), PEmag will pop up a dialog to ask whether to open the existing Maxwell2D project.



  a. If you select **Yes**, the Maxwell 2D project is reloaded from disk, and any changes made in Maxwell to the Maxwell 2D design and project are lost.
  b. If you select **No**, the Maxwell design and project on the Maxwell desktop will be overwritten by the design and project generated through PEmag with the latest changes in the PEmag model. The Maxwell 2D project is immediately saved to the disk.

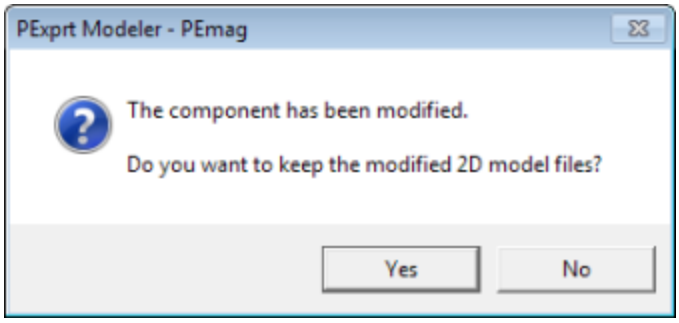
- If you launch PEmag from Maxwell with no Maxwell projects loaded; then open an existing PEmag project which has previously been used to generate a Maxwell project, modify the PEmag model and regenerate the Maxwell design (by selecting either **Modeler>FEA based Modeler(2D)>Maxwell 2D Link>Eddy Current** or **Modeler>FEA**

based Modeler(2D)>Maxwell 2D Link>Electrostatic), PEmag will pop up a dialog (see above) to ask whether to open the existing Maxwell 2D project.

a. If you click **Yes**, the Maxwell 2D project on disk is loaded to the Maxwell desktop.

b. If you click **No**, PEmag generates a Maxwell project and design on the Maxwell desktop, corresponding to the modified PEmag model. The project is immediately saved to the disk, overwriting the previous project on disk and deleting the previous solutions (if any).

- If you modify a PEmag model and regenerate the Maxwell design; then attempt to close PEmag, a dialog pops up asking whether to keep the modified 2D model files.



a. If you click **Yes**, PEmag closes and nothing is changed.

b. If you click **No**, PEmag closes, the Maxwell design and project is deleted from disk, but the modified Maxwell design remains on the Maxwell desktop, and you can save the project to disk, if desired. If not, closing Maxwell will permanently lose the project.

Related Topics

Creating Maxwell Designs Using PEmag

Solving Maxwell Designs Generated by PEmag

Solving Maxwell Designs Generated by PEmag

Maxwell designs generated by PEmag are fully setup and are ready for analysis.

- After solving the Maxwell design generated by PEmag, if you subsequently modify the PEmag model and regenerate the Maxwell design, the old Maxwell design is overwritten and the solutions are deleted.
- While the Maxwell design is being solved on the Maxwell desktop, if you modify the corresponding PEmag design on the PEmag desktop and invoke PEmag to regenerate the Maxwell design, the solver process on the Maxwell desktop will be aborted and the Maxwell design that has been modified by PEmag is loaded into Maxwell and saved to the disk simultaneously.

Related Topics

Creating Maxwell Designs Using PEmag

Modifying Maxwell Designs Using PEmag

Working with Variables

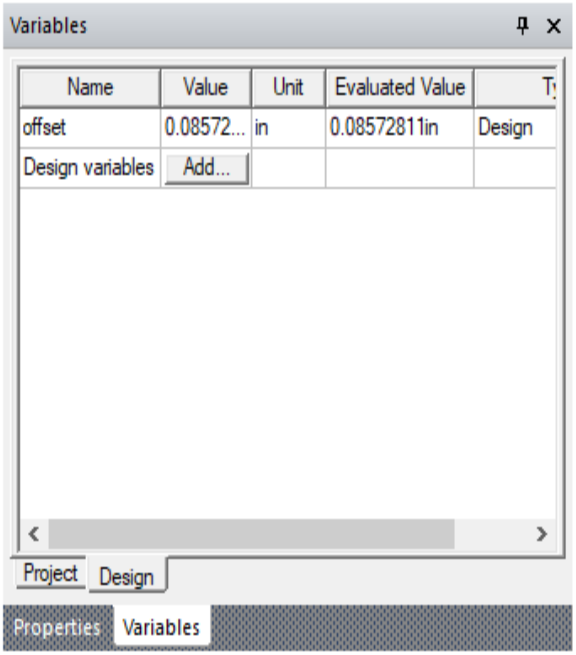
A variable is a numerical value, mathematical expression, or mathematical function that can be assigned to a design parameter in Maxwell. You can assign a variable to any dimensional or material property, or output value. Variables are useful in the following situations:

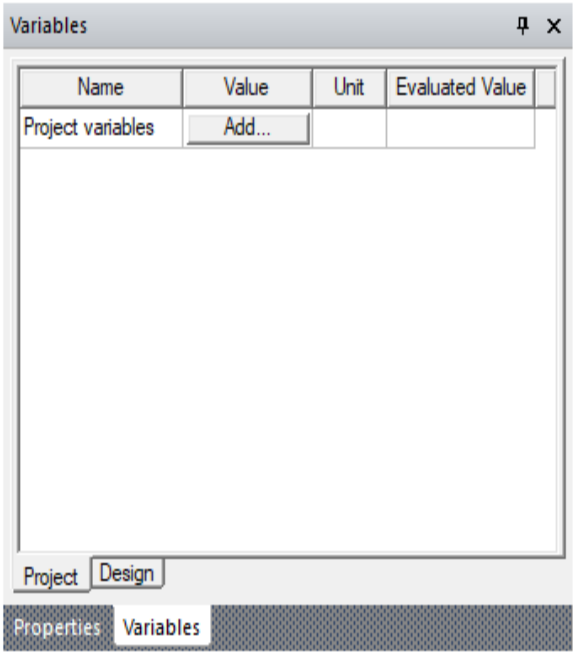
- When you expect to change a parameter often.
- When you expect to use the same parameter value often.
- When you intend to run a parametric analysis, in which you specify a series of variable values within a range to solve.
- When you intend to optimize a parameter value by running an optimization analysis.
- When you intend to run a convergence on an output variable.

There are two types of variables in Maxwell:

Project Variables	A project variable can be assigned to any parameter value in the project in which it was created. Maxwell differentiates project variables from other types of variables by prefixing the variable name with the **$** symbol. You can manually include the **$** symbol in the project variable's name, or Maxwell can automatically append the **$** after you define the variable. Project variables can be used in any design within the project.
Design Variables	A design variable can be assigned to any parameter value in the Maxwell design in which it was created. Design variables are not available to other designs within the same project.

Clicking **View** > **Variables** brings up a dockable variable window that is associated with the active project and/or design. When there is an active project, there will be a corresponding project variable tab. When there is an active design, there will be a corresponding design variable tab. Each tab contains an **Add...** button allowing creation of new variable of this type. If variables exist for the Project or Design, they are shown in the corresponding tab.





Related Topics

Defining an Expression
Defining Mathematical Functions
Assigning Variables
Viewing Output Variable Convergence
Specifying Output Variables
Using Optimetrics for Design Analysis
Choosing a Variable to Optimize
Selecting Objects by Variable
Exporting Variables for Documentation
Auto-Complete for Variables and Properties in Electronics Desktop

Adding a Project Variable

A project variable can be assigned to a parameter value in the Maxwell project in which it was created. Maxwell differentiates project variables from other types of variables by prefixing the variable name with the following symbol: **$**. You can manually include the symbol **$** in the project variable's name when you create it, or Maxwell automatically appends the project variable's name with the symbol after you define the variable. Project variables can be used in any design within the project.

1. Click **Project**>**Project Variables**.

- Alternatively, right-click the project name in the project tree, and then click **Project Variables** on the shortcut menu.

  You can also access the Project variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**.

The **Properties** dialog box appears.

2. Under the **Project Variables** tab, click **Add**.

   The **Add Property** dialog box appears.

3. In the **Name** box, type the name of the variable.

   Project variable names must start with the symbol **$** followed by a letter. Variable names may include alphanumeric characters and underscores (_). The names of intrinsic functions and the predefined constant pi (π) cannot be used as variable names.

   You can sort the project variables by clicking on the Name column header. By default, variables are sorted in original order. Clicking once sorts them in ascending order, noted by a triangle pointing up. Clicking against sorts in descending order, noted by a triangle pointing down. Clicking a third time sorts in original order, with no triangle.

4. Select a radio button for the variable use:

Selected Use	Setable Properties
Variable	Unit Type, Units, Value.
Separator	Value
Array Index Variable	Associate Array variable, Value

   Each selection affects the settable options.

5. For Project Variables in the **Unit Type** text box you can use the drop-down menu to select from the list of available unit types. “None” is the default.

   When you select a Unit Type, the choices in drop-down menu for the Units text box adapt to that unit type. For example, selecting Length as the Unit Type causes the Unit menu to show a range of metric and english units for length. Similarly, if you select the Unit Type as Resistance, the Units drop down lists a range of standard Ohm units.

6. In the **Value** text box, type the quantity that the variable represents. If you did not specify the Unit Type and Unit, you can also optionally, include the units of measurement.

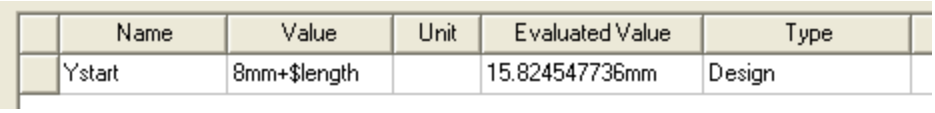
Warning	If you include the variable’s units in its definition (in the **Value** text box), do not include the variable’s units when you enter the variable name for a parameter value.

   The quantity can be a numerical value, a mathematical expression, or a mathematical function. The quantity entered will be the *current,* (or *default) value* for the variable. If the mathematical expression includes a reference to an existing variable, this variable is treated as a dependent variable. The units for a dependent variable will automatically change to

those of the independent variable on which the value depends. Additionally, dependent variables, though useful in many situations, cannot be the direct subject of optimization, sensitivity analysis, tuning, or statistical analysis.

7. Click **OK**.

   You return to the **Properties** dialog box. The new variable and its value are listed in the table. If the value is an expression, the evaluated value is shown. Updating the expression also changes the evaluated value display. The evaluated values of any dependent variables also are changed.



8. Optionally, type a description of the variable in the **Description** box.
9. Optionally, select **Read-Only**. The variable's name, value, unit, and description cannot be modified when **Read-only** is selected.
10. Optionally, select **Hidden**. If you clear the **Show Hidden** option, the hidden variable does not appear in the **Properties** dialog box.
11. You can also designate a variable as Sweep. You may need to use the scroll bar or resize the dialog to view the check boxes.

   The Sweep check box lets you designate variables to include in solution indexing as a way to permit faster post processing. Variables with Sweep unchecked are not used in solution indexing. If a solution exists, checking or unchecking a variable's Sweep setting produces a warning that the change will invalidate existing solutions. To continue, click **OK** to dismiss the warning dialog.

   If a variable has dependent variables, the Sweep check box is disabled and cannot be changed.

12. Click **OK**.

The new variable can now be assigned to a parameter value in the project in which it was created.

Related Topics

Adding a Design Variable
Deleting Project Variables
Defining an Expression
Defining Mathematical Functions
Exporting Variables for Documentation

Deleting Project Variables

To delete a project variable:

1. Remove all references to the variable in the project, including dependent variables.
2. **Save** the project to erase the command history.
3. Click **Project>Project Variables** to display the **Properties** dialog with list of variables. You can also access the Project variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**.
4. Select the variable and click **Remove** and **OK**.

To remove all unused project variables that are not in use (that is, not in the undo/redo command history):

1. Remove all references to unused project variables, including dependent variables.
2. **Save** the project to erase the command history.
3. Click **Project>Project Variables** to display the **Properties** dialog with list of variables. You can also access the Project variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**.
4. Click **Remove** and from the drop-down menu, select **Remove All Unused** and click **OK**.

   All variables that are not in use (not in undo/redo history) are removed.

To force remove all unused project variables:

1. Remove all references to unused project variables, including dependent variables.
2. Click **Project>Project Variables** to display the **Properties** dialog with list of variables. You can also access the Project variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**.
3. Click **Remove** and from the drop-down menu, select **Force Remove All Unused** and click **OK**. A warning message asks whether you want the unapplied changes in the property dialog to be applied, and clear undo/redo history. If you select No, nothing happens. If you

select Yes, all unapplied changes are applied first, and undo/redo history is cleared; all variables that are not used are removed (including those that are only in undo/redo history before this command is executed).

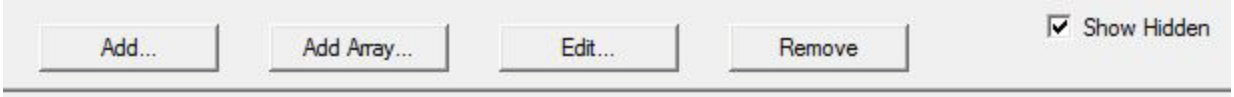
Adding a Design Variable

A design variable is associated with a Maxwell or RMxprt design. A design variable can be assigned to a parameter value in the design in which it was created. Design variables are only available within the design where they are created and are not available to other designs within the same project.

1. Click **Maxwell3D, Maxwell2D, or RMxprt, and then select Design Properties**.
   - Alternatively, right-click the design name in the project tree, and then click **Design Properties** on the shortcut menu.

     You can also access the Design variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Design Variables**.

   The **Properties** dialog box appears opened on the **Local Variables** tab. From the Properties dialog you can Add, Add Array, Edit, or Remove variables. This section describes **Add**. Use the links for descriptions of Add Array, Edit and Remove,



   Any existing Design variables are listed in the **Properties** dialog with the name followed by cells for Value, Unit, Evaluated Value, Type, Description, and Read-only and Hidden check boxes. A Show Hidden check box on the lower right of the Properties dialog controls the appearance of any Hidden variables.

   Initially, leave the radio button with Value selected until you have defined a variable. The other radio buttons let you enable defined variables for Optimization/Design of Experiments, Tuning, Sensitivity or Statistics. Selecting one of these radio buttons adds a new column to the Variable definition row for which you can check or uncheck Include for regular variables for that kind of Optimetrics simulation. For further discussion, see Optimetrics.

2. Click **Add**.

   The **Add Property** dialog box appears.

3. In the **Name** box, type the name of the variable.

   Variable names must start with a letter, and may include alphanumeric characters and underscores (_). The names of intrinsic functions and the predefined constant pi (π) cannot be used as variable names.

   You can sort the variables by clicking on the Name column header. By default, variables are sorted in original order. Clicking once sorts them in ascending order, noted by a triangle

pointing up. Clicking against sorts in descending order, noted by a triangle pointing down. Clicking a third time sorts in original order, with no triangle.

4. Select a radio button for the variable use:

Selected Use	Setable Properties
Variable	Unit Type, Units, Value
Array Index Variable	Associate Array variable, Value
Separator	Value
Post Processing Variable	Unit Type, Units, Value

Each selection affects the settable options.

5. In the **Unit Type** text box you can use the drop down menu to select from the list of available unit types. “None” is the default.

   When you select a Unit Type, the choices in drop down menu for the Units text box adapt to that unit type. For example, selecting Length as the Unit Type causes the Unit menu to show a range of metric and english units for length. Similarly, if you select the Unit Type as Resistance, the Units drop down lists a range of standard Ohm units.

6. In the **Value** box, type the quantity that the variable represents. Optionally, include the units of measurement.

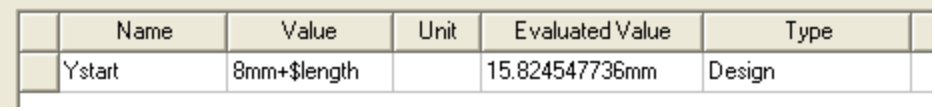
Note	If you include the variable's units in its definition (in the **Value** box), do not include the variable's units when you enter the variable name for a parameter value.

   The quantity can be a numerical value, a mathematical expression, or a mathematical function. The quantity entered will be the *current* (or *default value)* for the variable. If the mathematical expression includes a reference to an existing variable, this variable is treated as a dependent variable. The units for a dependent variable will automatically change to those of the independent variable on which the value depends. Additionally, dependent variables, though useful in many situations, cannot be the direct subject of optimization, sensitivity analysis, tuning, or statistical analysis.

Note	Complex numbers are not allowed for variables to be used in an Optimetrics sweep, or for optimization, statistical, sensitivity or tuning setups.

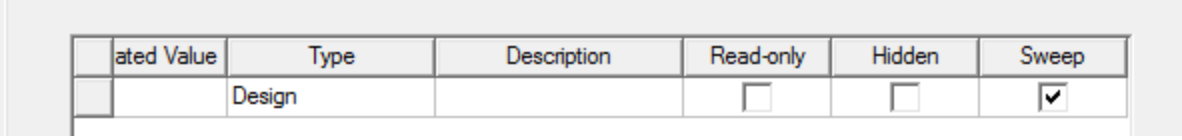
7. Click **OK**.

   You return to the **Properties** dialog box. The new variable and its value are listed in the table. If the value is an expression, the evaluated value is shown. Updating the expression also changes the evaluated value display. The evaluated values of any dependent variables also are changed.



	Name	Value	Unit	Evaluated Value	Type
	Ystart	8mm+$length		15.824547736mm	Design

8. Optionally, type a description of the variable in the **Description** box.
9. You can designate a variable as Read-only, Hidden, or Sweep. You may need to use the scroll bar or resize the dialog to view the check boxes.



The Sweep check box lets you designate variables to include in solution indexing as a way to permit faster post processing. Variables with Sweep unchecked are not used in solution indexing. If a solution exists, checking or unchecking a variable's Sweep setting produces a warning that the change will invalidate existing solutions. To continue, click **OK** to dismiss the warning dialog.

If a variable has dependent variables, the Sweep check box is disabled and cannot be changed.

10. Click **OK**.

    The new variable can now be assigned to a parameter value in the design in which it was created.

Related Topics

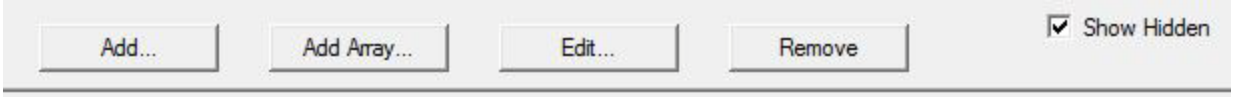
Adding a Project Variable
Deleting Design Variables
Deleting Project Variables
Defining an Expression
Defining Mathematical Functions

Add Array of Values for a Design Variable

A design variable is associated with a Maxwell design. You can also add a variable defined with an array of values.

1. Click **Maxwell 3D** or **Maxwell 2D**>**Design Properties**.
   - Alternatively, right-click the design name in the project tree, and then click **Design Properties** on the shortcut menu.

The Local Variables **Properties** dialog box appears. From the Properties dialog you can Add, Add Array, Edit... or Remove variables. This section describes Add Array. Use the links for descriptions of Add,.Edit and Remove,



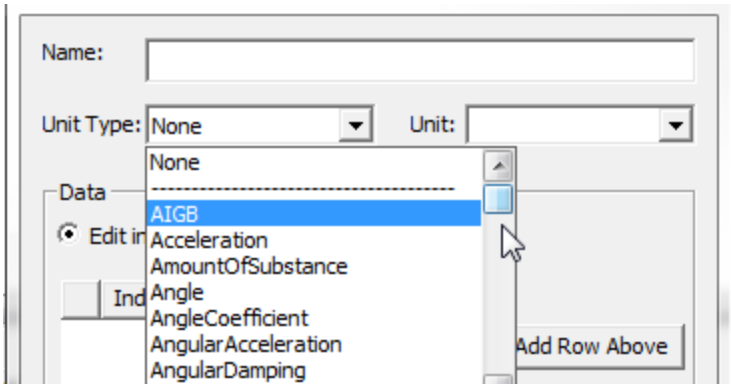
Any existing Design variables are listed in the **Properties** dialog with the name followed by cells for Value, Unit, Evaluated Value, Type, Description, and Read-only and Hidden check boxes. A Show Hidden check box on the lower right of the Properties dialog controls the appearance of any Hidden variables.

Initially, leave the radio button with Value selected until you have defined a variable. The other radio buttons let you enable regular variables for Optimization/Design of Experiments, Tuning, Sensitivity or Statistics. Array variables cannot be enabled for Optimetrics use.

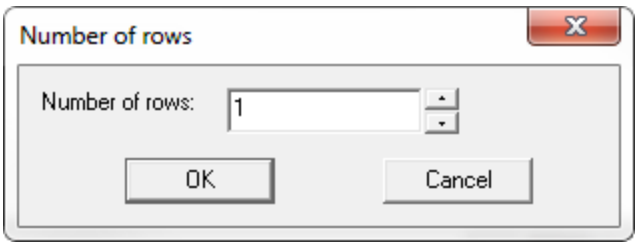
2. Click the **Add Array...**. button.

   The **Add Array** dialog displays.

3. Specify a variable Name in the text field.
4. Select a Unit Type and Units from the drop-down menus.



5. To specify the array with Edit in Grid Selected, you can begin by clicking the **Append Rows...** button to display the **Number of Rows** dialog box. (For Edit in plain text field, see below.)



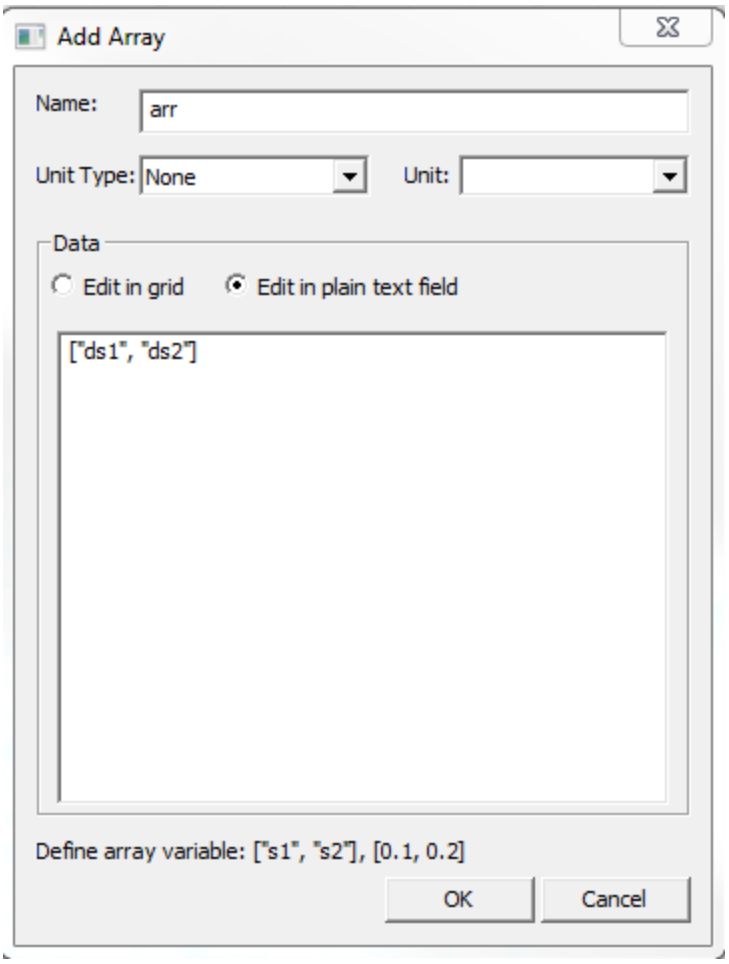
6. Specify a value and click OK.

This displays a list of indexed data rows in the **Add Array** dialog box. You can type any data value in the cells. If you enter alphanumeric text in a cell it must be delimited by double quotes. You can edit the rows relative a row selection by clicking buttons to Add Row Above, Add Row Below, or Remove Row. All cells must contain a value.

7. When you have completed the array, click **OK** to close the dialog,

   The Array variable is listed in the **Design Properties** dialog as a Local Variable. The array variable value field includes the array contents in brackets with the unindexed data values delimited by commas.

   If you elected to edit the array Edit in plain text field in the **Add Array** dialog, the bracketed and comma delimited format is used.



Related Topics

Adding a Project Variable

Deleting Design Variables

Deleting Project Variables

Defining an Expression

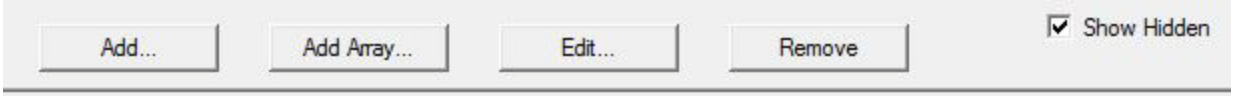
Defining Mathematical Functions

Editing a Design Variable

A design variable is associated with a Maxwell design. A design variable can be assigned to a parameter value in the Maxwell design in which it was created. You can also add a variable defined with an array of values.

1. Click **Maxwell**>**Design Properties**.
   - Alternatively, right-click the design name in the project tree, and then click **Design Properties** on the shortcut menu. You can also access the Design variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Design Variables**.

   The Local Variables **Properties** dialog box appears. From the Properties dialog you can Add, Add Array, Edit, or Remove variables. This section describes **Edit**. Use the links for descriptions of Add, Add Array and Remove.



   Any existing Design variables are listed in the **Properties** dialog with the name followed by cells for Value, Unit, Evaluated Value, Type, Description, and Read-only and Hidden check boxes. A Show Hidden check box on the lower right of the Properties dialog controls the appearance of any Hidden variables.

   Initially, leave the radio button with Value selected until you have defined a variable. The other radio buttons let you enable defined variables for Optimization/Design of Experiments, Tuning, Sensitivity or Statistics. Selecting one of these radio buttons add a new column to the Variable definition row for which you can check or uncheck Include a regular variable for that kind of Optimetrics simulation. Array variables cannot be Enabled for Optimetrics. For further discussion, see Optimetrics.
2. Select a variable to highlight it.
3. Click the **Edit** button.

   For regular variables, the **Add Property** dialog for that variable opens, and for array variables, the **Edit Array** variable dialog for that variable opens.
4. Complete the edits and **OK** the dialog to apply them.

Deleting Design Variables

To delete a design variable:

1. Remove all references to the variable in the design, including dependent variables.
2. **Save** the project to erase the command history.
3. Click **Maxwell>Design Properties** to display the **Properties** dialog with list of local variables.
4. Select the variable and click **Remove** and **OK**.

To remove all unused design variables (that is, not in the undo/redo command history):

1. Remove all references to unused variables in the design, including dependent variables.
2. **Save** the project to erase the command history.
3. Click **Maxwell 2D** or **Maxwell 3D>Design Properties** to display the **Properties** dialog with list of local variables.
4. Select the variable and click **Remove** and from the drop-down menu, select **Remove All Unused** and click **OK**. All variables that are not in use (not in undo/redo history) are removed.

To force remove all unused design variables:

1. Remove all references to unused variables in the design, including dependent variables.
2. Click **Maxwell 2D** or **Maxwell 3D>Design Properties** to display the **Properties** dialog with list of local variables.
3. Select the variable and click **Remove** and from the drop-down menu, select **Force Remove All Unused** and click **OK**. A warning message asks whether you want the unapplied changes in property dialog to be applied, and clear undo/redo history. If you select No, nothing happens. If you select Yes, all unapplied changes are applied first, and undo/redo history is cleared; all variables that are not used are removed (including those that are only in undo/redo history before this command is executed).

Related Topics

Deleting Project Variables

Adding Datasets

Datasets are collections of plotted data points that can be extrapolated into an equation based on the piecewise linear makeup of the plot. Each plot consists of straight line segments whose vertices represent their end points. A curve is fitted to the segments of the plot and an expression is derived from the curve that best fits the segmented plot. The created expression is then used in a piecewise linear intrinsic function such as: pwl, pwlx or pwl_periodic. You can add datasets at either the Project Level or the Design level. They can be for various purposes, including to define frequency dependent port impedances or frequency dependent global variables, and boundary definitions.

Project level datasets are used typically for defining various material properties at the project level (applicable to all designs in the project). Otherwise, project level datasets are not supported for design level assignment.

Note	The descriptions of the various controls and fields for working with datasets are also applicable to the **BH Curve** dialog box when *Specifying BH Curves for Nonlinear Relative Permeability*.

Design level datasets can be used in geometry entities like part commands, coordinate systems, points, and planes. Design level datasets do not work with equation-based surfaces or curves. Design level datasets can be used directly with piecewise linear functions in expressions or indirectly through variables that can refer to the dataset.

Design level datasets can be used in the following operations.

- **Creating or editing geometry** – when a geometry uses a dataset directly, edit dataset invalidates the solution. When a geometry uses a variable that is defined by dataset, editing the dataset does not invalidate the solution.
- **Creating an Animation** – based on a variable which can index in datasets.
- **Copying and Pasting Geometry** – if a part refers to a design dataset, it will be pasted to destination design.

The following procedure describes how to add a dataset manually. You can also add datasets:

- By importing data from an external file.
- By using the SheetScan tool.

1. For a Project level dataset, click **Project>Datasets**. For a Design level dataset, click **Maxwell 2D>Design Datasets**, **Maxwell 3D>Design Datasets**, or **RMxprt>Design Datasets**.

   The Datasets dialog box appears, listing any project or design-level datasets.

2. Click **Add**.

Note	If you are adding either a project-level or a design-level dataset, you will be presented with the option to add either a 1D Dataset or 3D Dataset. See: Adding 3D Datasets for specifics on this type of dataset.

   The **Add Dataset** dialog box appears. The dialog contains fields for the Dataset name, and a table for x- and y- coordinates. It contains a graphic display that draws a line for the coordinates you add. It also includes buttons for the following functions:

   - **Swap X-Y Data** – this swaps the x- and Y- coordinates and adjusts the graphical display.
   - **Import Dataset** - this provides a way to import data sets from an external source. The format is a tab separated points file. Clicking the button opens a file browser window.
   - **Export Dataset** -- this provides a way to export the current dataset to a tab separated points file. Clicking the button opens a file browser window.
   - **Add Row Above** – adds a new row to the table above the selected row.
   - **Add Row Below** – adds a new row to the table below the selected row/
   - **Append Row** – opens a dialog that lets you specify a number of rows to add to the table.
   - **Delete Row** – deletes the selected row or rows.

3. Optionally, enter a name other than the default for the dataset in the **Name** box.

Note	Project-level datasets must begin with a dollar sign character ($). If you do not type one, it will be added for you.

4. Enter the x- and y- coordinates by one of the following methods

   Type the x- and y-coordinates for the first data point in the row labeled **1**.Type the x- and y-coordinates for the remaining data points in the dataset using the same method.

   After you type a point's coordinates and move to the next row, the point is added to the plot, adjusting the display with each newly entered point.

Note	• The x-coordinate values for successive data points must increase within ten significant digits. • You can also use the **Import Dataset** button to import data coordinates. Doing so will overwrite any existing values in the Coordinates panel.

5. When you are finished entering the data point coordinates, click **OK**.
6. Click **Done** in the **Datasets** dialog box.

The dataset plot is extrapolated into an expression that can be used in parametric analyses, boundary definitions, or assigned to a material property value.

You can now edit, remove, clone, or export the dataset. You can also change the plot display properties.

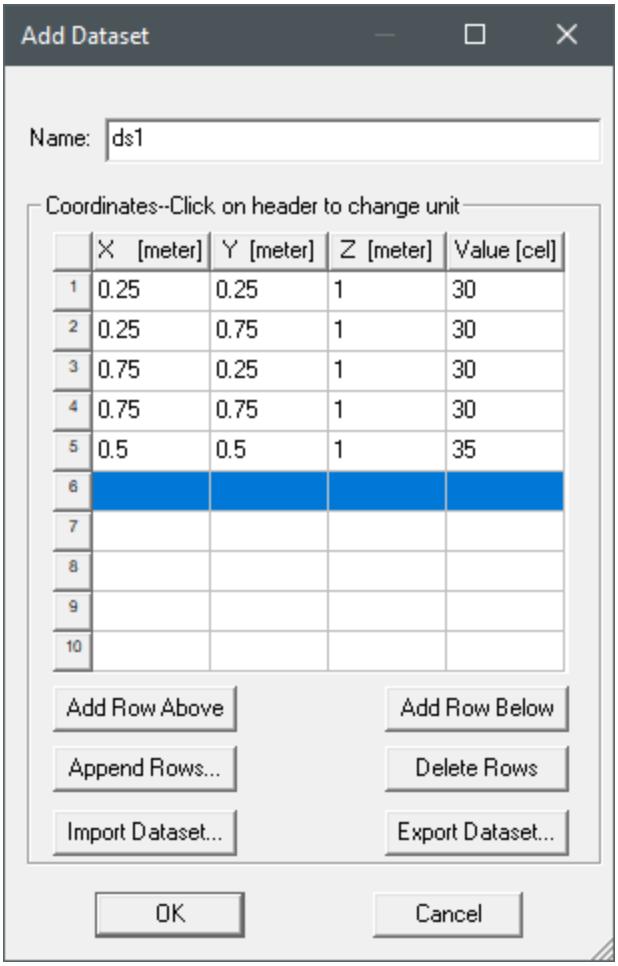
Related Topics

Adding a Design Variable
Adding 3D Datasets
Editing Datasets
Using Piecewise Linear Functions in Expressions
Using Dataset Expressions
Calculating the Properties for a Non-Linear Permanent Magnet
Using SheetScan

Adding 3D Datasets

In Maxwell designs, you can define both project-level and design-level 3D datasets. Use a project-level 3D dataset to define material properties that vary according to the spatial coordinates. Use a design-level 3D dataset to define temperatures that vary according to the spatial coordinates.

To define a 3D dataset, specify a list of **X**, **Y**, and **Z** coordinates and the corresponding **Values** in the **Add Dataset** window. For example, the first line in the image below specifies a temperature of 30° C at (0.25, 0.25, 1). The data specifies a variation of temperature on the plane $0.25 \le X \le 0.75$, $0.25 \le Y \le 0.75$, $Z = 1$. In this example, the values in the **Values** column are temperature, but can be set to a different unit depending on the property you wish to assign. To change the unit, click the **X**, **Y**, **Z**, or **Value** header row.



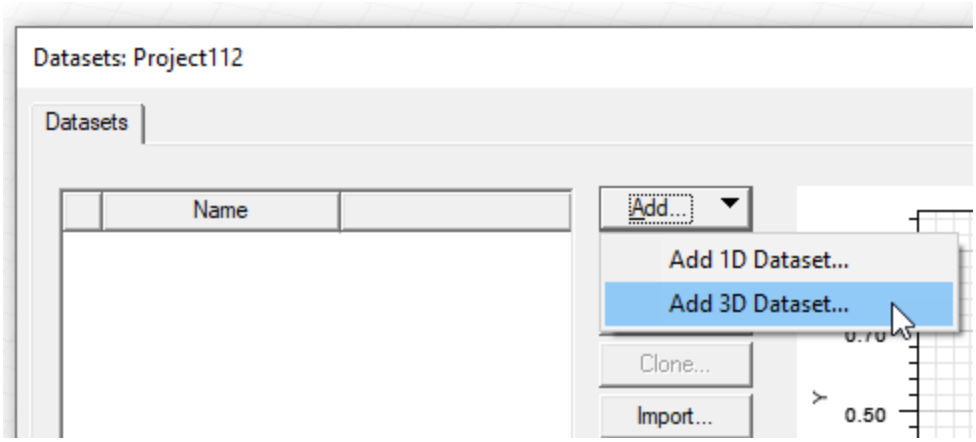
1. Depending on whether you are defining a project-level or design-level 3D dataset, choose the appropriate one of the following two options:
   - For project-level datasets (material properties):

     Using the menu bar, click **Project** > **Datasets** or, on the **Desktop** ribbon tab, click **Datasets**.
   - For design-level datasets (temperatures):

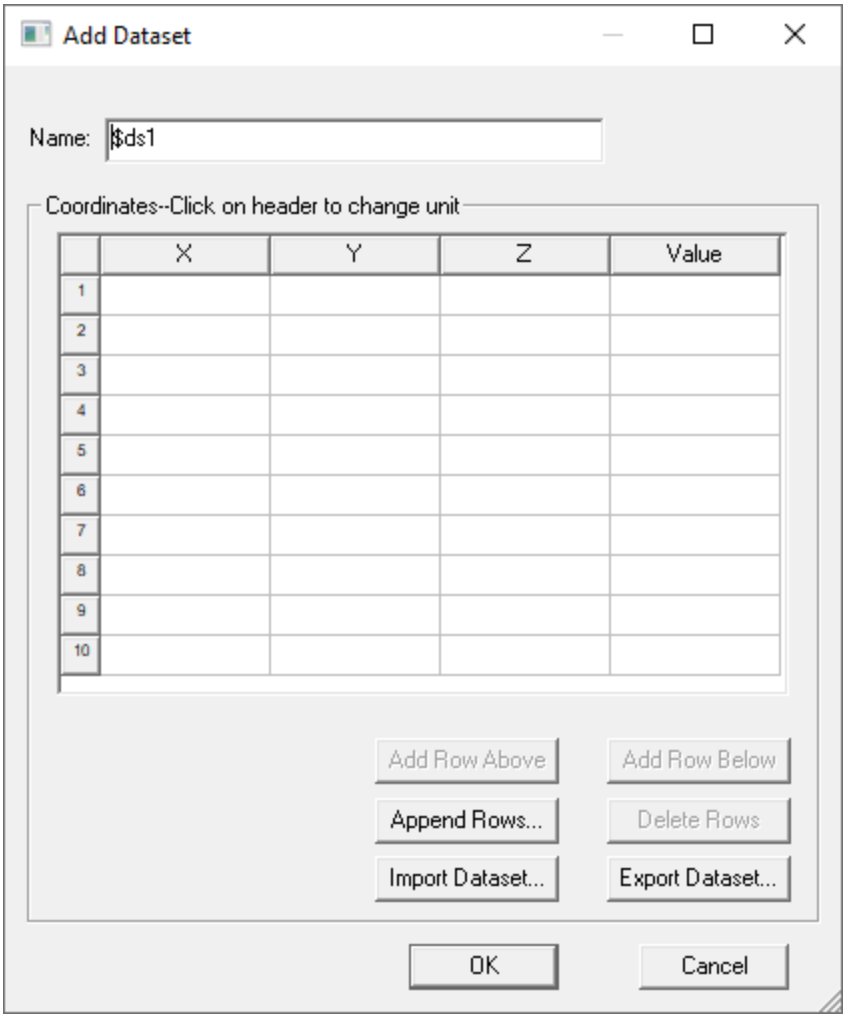
     Using the menu bar, click **Maxwell** > **Design Datasets**.

   The **Datasets** dialog box appears. This lists any existing datasets.

2. Click **Add** > **Add 3D Dataset**.



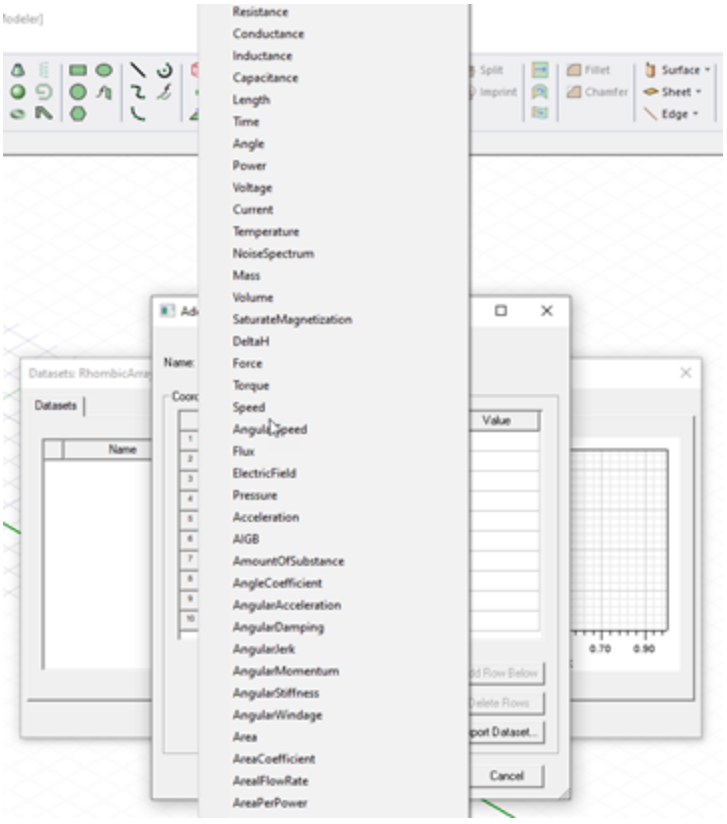
The **Add Dataset** window appears.



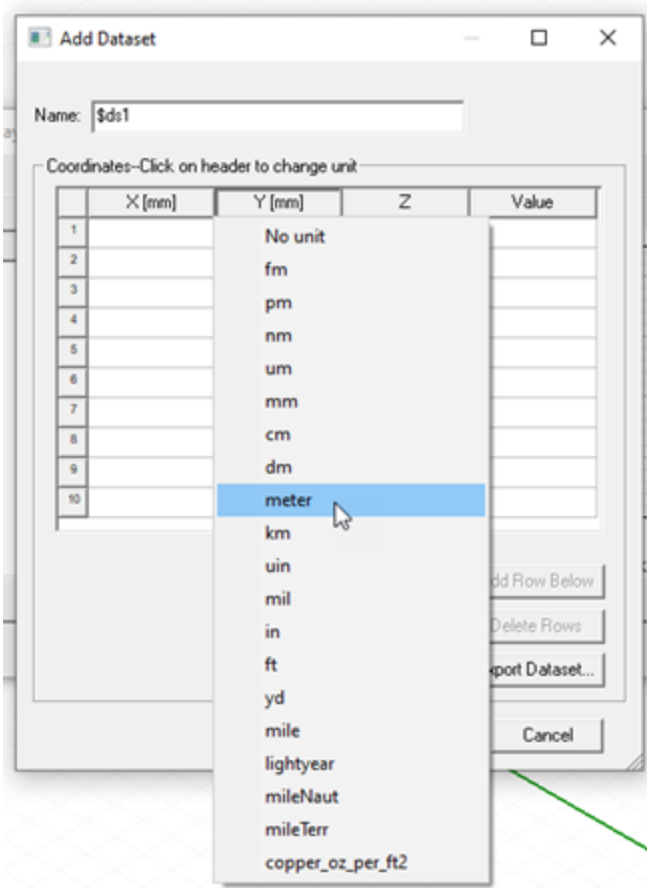
The window contains fields for the dataset name and a table for X, Y, and Z coordinates and corresponding values. It can be resized to display additional rows and columns, and includes buttons for the following functions:

- **Add Row Above** – adds a new row to the table above the selected row.
- **Add Row Below** – adds a new row to the table below the selected row.
- **Append Row** – opens a dialog box that lets you specify a number of rows to add to the table.
- **Delete Row** – deletes the selected row or rows.
- **Import Dataset** – this provides a way to import data sets from an external source. The format is a tab separated points file. Clicking the button opens a file browser window.
- **Export Dataset** – this provides a way to export the current dataset to a tab separated points file. Clicking the button opens a file browser window.

3. Optionally, type a name other than the default for the dataset in the **Name** field.
4. Click the **X**, **Y**, **Z** and **Value** column headers to select a unit category. The default is No Unit.



Click the header again to select a unit in that category. You can select No unit.



Selected units are displayed for each column in the header cell.

5. Enter the X, Y, and Z coordinates and the corresponding values by one of the following methods:
   - **Import Dataset**
   - Type coordinates and associated value in the **X**, **Y**, **Z**, and **Value** columns.
6. When you are finished entering the data point coordinates and values, click **OK**.

Note	When used with a clp function (Closest point interpolation) in an expression, performance will degrade with large 3D datasets.

Datasets Dialog Box

The **Datasets** dialog box provides a browsable listing of all datasets currently defined for the project or design. A preview window displays a plot of the currently selected dataset. Controls allow you to **Add**, **Edit**, **Remove**, and **Clone** datasets; to **Import** and **Export** characteristics data; and to launch the **SheetScan** tool that you can use to extract data from graphics such as data sheets.

Add - Opens a blank **Add Dataset** window in which you can define a dataset by entering data coordinates directly, or by importing data from a file.

Edit - Opens the selected dataset in an **Edit Dataset** window for editing.

Remove - Removes the selected dataset from the project.

Clone - Copies the selected dataset to an editing window for modification. The original dataset remains intact.

Import - Opens an **Import Dataset** window in which you can locate and import characteristics data from several file types including data exported from the **SheetScan** tool.

Export - Opens an **Export Dataset** window in which you can export the selected dataset to a tab-delimited file.

SheetScan - Opens the SheetScan tool in which you can extract characteristics data from graphics such as data sheets.

Related Topics

Adding Datasets
Importing Datasets
Editing Datasets
Cloning Datasets
Exporting Datasets
Removing Datasets
Using SheetScan

Dataset Preview Plot Properties

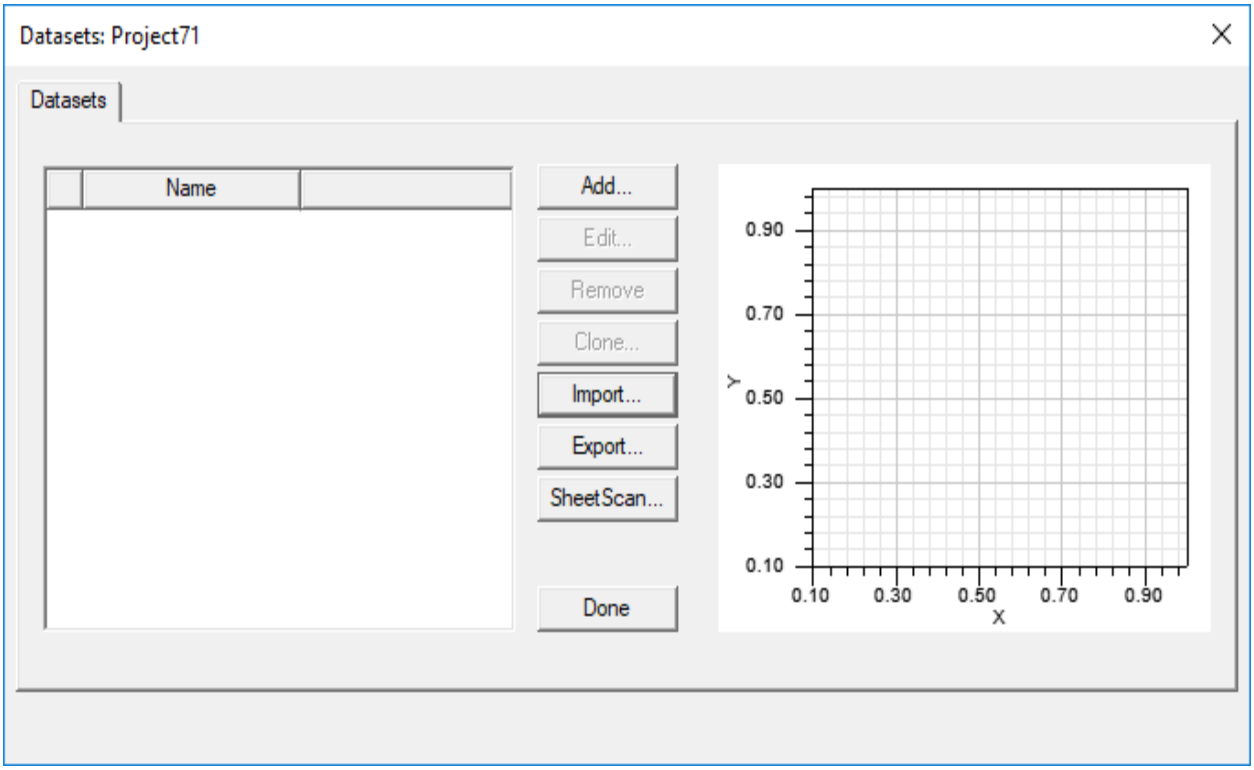
1. You can change the properties of various elements of the preview plot such as the major and minor grid colors, title font, and trace line style by double-clicking the element to open its properties dialog box.
2. Depending on the type of element selected, tabs allow you to set the **Color**, **Font**, **Line Style**, **Scaling**, **Title**, and **Legend** properties.
3. Right-clicking inside the plot window opens a context menu containing commands to add and delete data markers and labels, change trace type, and print the plot.
4. Click **Save as Defaults** to make the changes the default values.
5. Click **OK** to apply the changes only while the current dataset plot window is open. Click Save as Defaults to keep the changes.

Importing Datasets

To import data for a dataset from a file:

1. For a Project level dataset, click **Project**>**Datasets**. For a Design level dataset, click **Maxwell 2D>Design Datasets**, **Maxwell 3D>Design Datasets**, or **RMxprt>Design Datasets**.

The **Datasets** dialog box appears.



2. Click the **Import** button.
3. In the file browser window that appears, choose the data file you wish to import from the following file types:

Note	The table below lists the file types supported for *Project level datasets*. For *Design level datasets*, only **.tab** files are supported.

.mdx, .mda	Twin Builder Characteristic format
.xls, .xlsx	Microsoft Excel
.txt	text file
.csv	Comma-separated value
.out	Maxwell SPICE (read-only - reads data inside the KW_DATA section)
.cfg	Comtrade (IEEE Std C37-111-1999)
.dat	TEK Oscilloscope
.tab	tab delimited data files

- If you select a file type other than .**xls**, .**xlsx**, .**txt**, .**dat**, .**out**, or .**csv** the data is imported immediately into the **Add Dataset** dialog box.

Note	Importing an .xls or .xlsx file requires Microsoft Excel to be available.

- Selecting an .xls or .xlsx file containing multiple sheets opens a **Table Properties** dialog box where you can choose the desired sheet from a drop-down list. Otherwise, selecting an .xls or .xlsx file imports the data immediately into the **Add Dataset** dialog box.

Note	Only the first two columns of data are imported, the left-most column containing the X-coordinate values. The x-coordinate values for successive data points must increase within ten significant digits. Non-numeric entries are assigned a value of zero. The first row of data is assumed to contain column headings and is ignored.

- Selecting a .**txt** or .**csv** file opens an **Import** dialog box in which you can specify how to settings for reading the data in the file for import. You can choose the **Separator**(s) and **Decimal Symbol**, as well as the line at which to begin the import. The dialog box shows both the original text and the text as it would appear when imported based on the current import settings.

When satisfied with the import settings, click **OK** to import the data.

4. After importing the data, you can modify it manually (see Editing Datasets).

Note	You can also import a dataset in the **Add Dataset** and **Edit Dataset** windows.

Related Topics

Adding Datasets
Editing Datasets
Cloning Datasets
Exporting Datasets
Removing Datasets
Using SheetScan

Editing Datasets

1. For Project level datasets, click **Project>Datasets**. For Design level datasets, click **Maxwell 2D>Design Datasets**, **Maxwell 3D>Design Datasets**, or **RMxprt>Design Datasets**.

   The **Datasets** dialog box appears.
2. Click the dataset name you want to modify, and then click **Edit**.

   The **Edit Dataset** dialog box appears.
3. Optionally, type a name other than the default for the dataset in the **Name** box.
4. Type new values for the data points as desired.

   The plot is adjusted to reflect the revised data points.

5. When you are finished entering the data point coordinates, click **OK**.
6. Click **Done.**

Related Topics

Adding Datasets
Adding a Design Variable
Modifying Datasets
Using Piecewise Linear Functions in Expressions
Using Dataset Expressions

Cloning Datasets

Cloning a dataset generates a copy of an existing dataset. The clone can then be modified as needed.

1. For Project level datasets, click **Project>Datasets**. For Design level datasets, click **Maxwell 2D>Design Datasets**, **Maxwell 3D>Design Datasets**, or **RMxprt>Design Datasets**.

   The **Datasets** dialog box appears.
2. Click the dataset name you want to clone, and then click **Clone**.

   The **Clone Dataset** dialog box appears.
3. Modify the dataset as needed.

Related Topics

Adding Datasets
Importing Datasets
Editing Datasets
Exporting Datasets
Removing Datasets
Using SheetScan

Exporting Datasets

1. For Project level datasets, click **Project>Datasets**. For Design level datasets, click **Maxwell 2D>Design Datasets**, **Maxwell 3D>Design Datasets**, or **RMxprt>Design Datasets**.

   The **Datasets** dialog box appears.
2. Click the **Export** button.

   The **Export Dataset** dialog box appears.

3. Browse to the location you want to store the exported dataset.
4. Name the file (exported datasets are tab-delimited and are given a .tab extension), and click **Save** to complete the export operation.

Note	You can also export a dataset in the Add Dataset and Edit Dataset windows.

Related Topics

Adding Datasets
Importing Datasets
Editing Datasets
Cloning Datasets
Removing Datasets
Using SheetScan

Removing Datasets

1. For Project level datasets, click **Project>Datasets**. For Design level datasets, click **Maxwell 2D>Design Datasets**, **Maxwell 3D>Design Datasets**, or **RMxprt>Design Datasets**.

   The **Datasets** dialog box appears.
2. Click the dataset name you want to remove, and then click **Remove**.
3. When finished removing datasets, click **Done**.

Related Topics

Adding Datasets
Importing Datasets
Editing Datasets
Cloning Datasets
Exporting Datasets
Using SheetScan

Using SheetScan

SheetScan allows you to extract characteristics data from graphics such as data sheets which have been scanned and saved in any of the following formats: .bmp, .dib, .jpg, .gif, .tif, .tga, .pcx, .htm, or .html. SheetScan can be accessed at the AEDT project level under **Project>Datasets>SheetScan**.

In addition to importing graphic files directly, SheetScan also can be used to browse the Internet for datasheet information and transfer a snapshot of the web page to the SheetScan editor where

you can map axes on the image as an overlay. You can then manually add datapoints to approximate the characteristic curve(s) on the datasheet.The sampled data can then be converted to Maxwell format, and the extracted data exported to a Maxwell dataset or saved to a tab-delimited file.

The process for creating a dataset using SheetScan involves four basic operations:

- Loading a datasheet into SheetScan.
- Defining a coordinate system for the imported datasheet picture.
- Defining a characteristic curve using the datasheet picture as reference.
- Exporting the characteristic curve data to a file or to a dataset.

Related Topics

SheetScan Toolbars

SheetScan Settings

The Curve Values Window

Loading a Datasheet into SheetScan

Deleting a Datasheet Picture

Defining a SheetScan Coordinate System

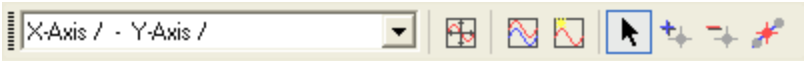
Defining a Characteristic Curve in SheetScan

SheetScan Toolbars

Three toolbars are available in SheetScan. They provide convenient access to commands that are also found in the SheetScan main menu. Toolbars can be toggled on and off via the **View>Toolbar** sub-menu.

- The **Standard** toolbar includes access to basic Windows functions such as file Open and Save, Cut, Copy, Paste, Print, and Help.

- The **Curve** toolbar contains tools for working with curve values. A pull-down menu allows you to select the curve on which to work. Other tools allow you to: change curve settings, change the curve's coordinate system, and to select, append, delete, and insert points on the active curve



- The **Zoom** toolbar provides tools for scaling the current view, zooming in and out, resetting

the zoom to 100 percent, and toggling the display of the curve's grid on and off.

Related Topics

SheetScan Settings
The Curve Values Window
Loading a Datasheet Picture into SheetScan
Deleting a Datasheet Picture
Defining a SheetScan Coordinate System
Defining a Characteristic Curve in SheetScan

SheetScan Settings

Default settings are made by choosing **Options>Settings**. The **Settings** dialog box contains three tabs:

- The **Document** tab allows you to set the Width and Height of the sheet created when a picture imported into the SheetScan editor. You can either enter the dimensions manually, or allow SheetScan to adapt the dimensions to the picture being loaded.
- The **Axis** tab allows you to set the default **Name**, **Unit** of measure, **Scaling** factor, and **Offset** value for the X and Y axes. Checking **Monotonicity in X** automatically prevents you from adding consecutive data points whose X-values are not increasing.
- The **Representation** tab lets you choose whether to connect points on the characteristic curve and to choose the color of the connecting line. You can also choose to display markers for the point chosen when defining a curve, to set the color of displayed markers, and to set the color of markers when they are selected.

Note	You can also override the default settings on the Axis and Representation tabs for individual curves (see Defining a Characteristic Curve in SheetScan).

Related Topics

Loading a Datasheet Picture into SheetScan
Defining a SheetScan Coordinate System
Defining a Characteristic Curve in SheetScan

The Curve Values Window

The **View>Curve Values** menu item toggles the display of a dock-able **Curve Values** window that displays the data points you place when creating a characteristic curve. Data for each curve

on a sheet is displayed on its own tab. You can manually change the X and Y values in the table to fine-tune the characteristic curve.

Related Topics

Loading a Datasheet Picture into SheetScan
Defining a SheetScan Coordinate System
Defining a Characteristic Curve in SheetScan

Loading a Datasheet Picture into SheetScan

By default, SheetScan opens a new, blank datasheet editing window. There are two ways to load a datasheet picture into the editor:

Loading a Datasheet Picture Directly

1. Browse directly to the datasheet picture file by choosing **Picture>Load picture** to open a file browser window.
2. When you have located the desired file, click **OK** to load the image into the SheetScan editor. Supported file types include: .bmp, .dib, .jpg, .gif, .tif, .tga, .pcx, .htm, and .html.

Loading a Datasheet Picture Using the HtmlViewer

1. Choose **Picture>Internet** to open the Sheetscan HtmlViewer.
2. Browse the Internet for the desired datasheet.
3. Resize the **HtmlViewer** window and adjust its scrollbars until the desired portion of the datasheet is in view
4. Click the **To SheetScan** button  to copy the visible contents of the HtmlViewer window into the SheetScan editor window.

Note	To hide the datasheet picture, choose **Picture** on the **View** menu.

After loading a datasheet picture into the SheetScan editor, the next step is Defining a coordinate system for the imported picture.

Related Topics

SheetScan Toolbars
SheetScan Settings
The Curve Values Window
Deleting a Datasheet Picture
Loading a Datasheet Picture into SheetScan
Defining a SheetScan Coordinate System
Defining a Characteristic Curve in SheetScan

Deleting a Datasheet Picture

To delete a Datasheet picture, choose **Picture>Delete picture**.

Warning	You cannot undo this action. If you delete a picture from the SheetScan editor, you must reload it from the source file or Internet web page.

Related Topics

Loading a Datasheet Picture into SheetScan

Defining a SheetScan Coordinate System

Use this procedure to define a coordiante system typically for a graph on a datasheet picture that you have previously loaded into the SheetScan editor.

To define the coordinate system:

1. Select **Coordinate System>New** to open the **Coordinate System** dialog box.
2. Click the **Point1** button.

   The **Coordinate System** dialog box disappears temporarily and the cursor changes to a crosshairs.
3. Position the cursor over a corner of the datasheet graph and click the left mouse button.

   The **Coordinate System** dialog box reappears displaying the X- and Y-Coordinate values for the chosen point.
4. Enter the X-Value and Y-Value for this point. Typically, these values will correspond to the values taken from the axis scale values on the datasheet.
5. Select the desired scaling (linear, logarithmic, or decibel) for both the X and Y axes.
6. Repeat steps 2 through 4 for the **Point2** and **Point3** buttons.
7. Click **OK**. The grid is placed in the graphic.

Note	• You can edit the grid after placement either by selecting **Coordinate System>Properties** from the SheetScan main menu bar, by clicking the coordinate system icon on the Curve toolbar, or by right-clicking in the SheetScan editing window and selecting **Coordinate system** from the context menu. • You can hide the grid by selecting **Grid** on the **View** menu, or by clicking the grid icon on the **Curve** toolbar.
8. Proceed to *Defining a Characteristic Curve in SheetScan*.

Related Topics

SheetScan Toolbars

SheetScan Settings

Loading a Datasheet Picture into SheetScan

Defining a Characteristic Curve in SheetScan

Defining a Characteristic Curve in SheetScan

Once you have loaded a datasheet picture in the editor and have defined a coordinate system, you can define one or more characteristic curves as follows:

1. Choose **Curve>New**.

   The **Curve Settings** dialog box opens.
2. Define the properties of the curve. Refer to SheetScan Settings for a detailed explanation of the settings you can make on the **Axis** and **Representation** tabs.
3. When finished defining curve properties, click **OK**.

   The cursor changes to cross hairs.
4. Click the points of the characteristic which you want to capture for the dataset. The points are connected automatically.
5. Repeat steps 1 through 4 for each additional characteristic curve you wish to define.

After characteristic curves have been defined, you can perform various operations on them. Refer to *Performing Operations on SheetScan Curves* for details.

Related Topics

SheetScan Toolbars

SheetScan Settings

The Curve Values Window

Performing Operations on SheetScan Curves

You can perform the following operations on existing characteristic curves:

- select a curve for editing
- edit a curve
- delete a curve
- change curve settings
- check if a curve is monotonically increasing along the X-axis
- export the curve data to a dataset

Selecting a SheetScan Characteristic Curve

1. To select a SheetScan curve for editing, do one of the following:
   a. Select **Curve>Select** on the SheetScan menu bar.
   b. Right-click in the editor window and choose **Select Curve** from the context menu.

The **Select Curve** dialog box opens.

2. Click the desired curve name to highlight it, then click **OK** to select the curve.

Note	If the Curve Values window is open, you can also click the tab of the desired curve to select it.

Related Topics

SheetScan Toolbars
SheetScan Settings
The Curve Values Window

Changing Characteristic Curve Settings

You can changes curve settings for a characteristic curve as follows:

1. Select the curve whose settings you wish to change.
2. Choose **Curve>Change Settings**.

   The **Curve Settings** dialog box opens.
2. Change the properties of the curve as desired. Refer to SheetScan Settings for a detailed explanation of the settings you can make on the **Axis** and **Representation** tabs.
3. When finished changing curve properties, click **OK**.
4. Repeat steps 1 through 3 for each additional characteristic curve you wish to change.

Related Topics

SheetScan Toolbars
SheetScan Settings
The Curve Values Window

Editing a SheetScan Characteristic Curve

The following SheetScan curve editing functions are available either via the main **Curve** menu, the Curve toolbar, or the editor window context menu:

- Select points — click on a point to select it. **Ctrl+click** selects multiple points.
- Append points — click to add data points to the end of a curve.
- Delete points — click a data point to remove it from the curve.
- Insert points — click to insert new data points between existing data points.

Related Topics

SheetScan Toolbars
SheetScan Settings

The Curve Values Window

Deleting a SheetScan Characteristic Curve

To delete a characteristic curve and all of its associated data points:

1. Select the desired curve.
2. Select **Curve>Delete**.

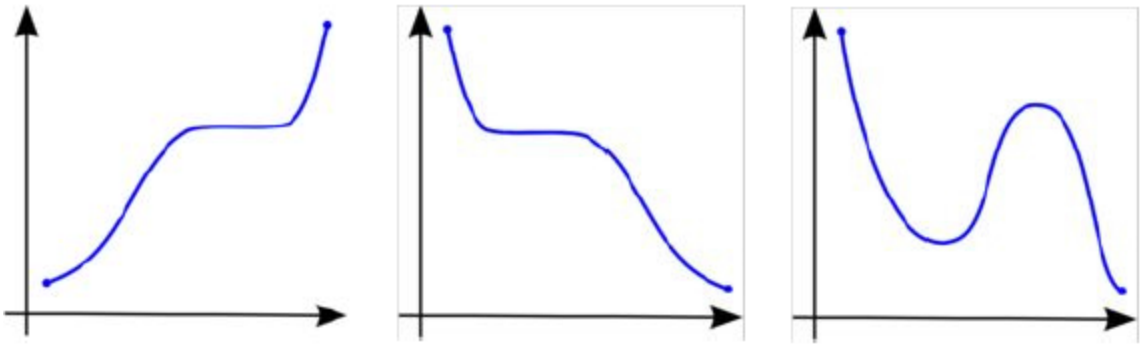
Warning	You cannot undo this action. If you delete a curve and its data points from the SheetScan editor, you must reconstruct it manually.

Checking for Monotonicity in X

Maxwell requires that characteristics be monotonically increasing along the X-axis. In other words, successive data points must have increasing X-values, while Y-values may both increase and decrease. You can check for monotonicity in X as follows:

1. Select **Curve>Check Monotonicity**.

   If the characteristic curve is monotonically increasing in X-value, the check completes without notice. Typical examples of curves that meet monotonicity criteria are shown below.



2. If the characteristic curve is not monotonically increasing in X-value, a dialog box displays informing you that errors were found. Click **Yes** to have SheetScan automatically correct the errors.

Related Topics

Defining a Characteristic Curve in SheetScan

Importing Characteristic Data into SheetScan

SheetScan supports data import from the following file types: Twin Builder Characteristic (*.mdx, *.mda), Microsoft Access (*.mdb), Microsoft Excel (*.xls), text (*.txt), comma separated value

(*.csv), Spice (*.out), Comtrade (*.cfg), and TEK Oscilloscope (*.dat).

1. To import characteristic curve data into SheetScan, select **File>Import**.
2. In the file **Open** dialog box, select the desired data file and click **OK**.
   a. Selecting a **.xls** file containing multiple sheets opens a **Table Properties** dialog where you can choose the desired sheet from a drop-down list. Otherwise, selecting a **.xls** file imports the data immediately into the **Add Dataset** dialog box.

Note	• Only the first two columns of data are imported, the left-most column containing the X-coordinate values. The x-coordinate values for successive data points must increase within ten significant digits. Non-numeric entries are assigned a value of zero. • The first row of data is assumed to contain column headings and is ignored.

   b. Selecting a **.txt**, **.tab**, or **.csv** file opens an **Import** dialog box in which you can specify how to settings for reading the data in the file for import. You can choose the **Separator** (s) and **Decimal Symbol**, as well as the line at which to begin the import. The dialog box shows both the original text and the text as it would appear when imported based on the current import settings.

   When satisfied with the import settings, click **OK** to import the data.

   A **Curve Settings** dialog box opens.

3. Change **Curve Settings** as needed and click OK to complete the data import.
   The new characteristic curve is added to the current SheetScan sheet.

Exporting SheetScan Data

You can export SheetScan curve data directly to a Maxwell or RMxprt Dataset.

1. To export the curve data, choose **File>Export**.
2. In the **Save**dialog box, choose **Current Curve** (default) to export current curve data, or **Curves** if you wish to choose the curve(s) whose data you wish to export. Choosing **Curves** reveals a list box showing all of the curves available for export. Check the **Export** box for the desired curves.

Note	The **Multidimensional Table** option is not currently supported.

3. Choose **Equidistant** if you want to set the **Start** and **StopX-Channel** values and a **Sample Rate** or **Number of samples** for the exported dataset(s),
4. Choose **Dataset** to export curve data directly to the project's dataset file.

Note	Exporting data to a file via the **File** button is not currently supported.

Related Topics

Adding Datasets

Defining Mathematical Functions

A mathematical function is an expression that references another defined variable. A function's definition can include both expressions and variables.

The following mathematical functions may be used to define expressions:

Basic functions	**/**, **+**, **-**, *****, mod (modulus), ****** (exponentiation), **-** (Unary minus), **==** (equals), **!** (not), **!=** (not equals), **>** (greater than), **<** (less than), **>=** (greater than equals), **<=** (less than equals), **&&** (logical and), **||** (logical or)
Intrinsic functions	**if**, **abs**, **exp**, **pow**, **ln** (natural log), **log10**, **sqrt**
Trigonometric expressions	**sin**, **cos**, **tan**, **asin**, **acos**, **atan**, **sinh**, **cosh**, **tanh**

The predefined variables X, Y, Z, Phi, Theta, R, and Rho must be entered as such. X, Y, and Z are the rectangular coordinates. Phi, Theta, and Rho are the spherical coordinates. R is the cylindrical radius, and Rho is the spherical radius.

If you do not specify units, all trigonometric expressions expect their arguments to be in radians, and the inverse trigonometric functions' return values are in radians. If you want to use degrees, you must supply the unit name **deg**. When the argument to a trigonometric expression is a variable, the units are assumed to be radians. These function names are reserved and may not be used as variable names.

As far as expression evaluation is concerned: units are conversion factors (that is, from the given unit to SI). Note also that the evaluated value of an expression) is always interpreted as in SI units.

Related Topics

Setting Coordinate Systems

Expressing Cartesian, Cylindrical or Spherical Coordinates

Defining an Expression

Expressions are mathematical descriptions that typically contain intrinsic functions, such as sin(x), and arithmetic operators, such as +, -, *, and /, well as defined variables. For example, you could define: x_size = 1mm, y_size = x_size + sin(x_size). Defining one variable in terms of another makes a dependent variable. Dependent variables, though useful in many situations, cannot be the subject of optimization, sensitivity analysis, tuning, or statistical analysis.

The Constants tab of the **Project Variables** dialog lists the available predefined constants.These may not be reassigned a new value.

Name	Value	Description
Boltz	1.3806503E-023	Boltzmann constant (J/K)
c0	299792458	Speed of light in vacuum (m/s)
e0	8.854187817e-012	Permittivity of vacuum (F/m)
elecq	1.602176462e-019	Electron charge (C)
eta	376.730313461	Impedance of vacuum (Ohm)
pi	3.14159265358979	Ratio of circle circumference to diameter
u0	1.25663706143582e-066	Permeability of vacuum (H/m)

Numerical values may be entered in Ansys' shorthand for scientific notation. For example, $5x10^7$ could be entered as **5e7**.

Note	Maxwell 2D, 3D, and RMxprt do not support expressions that include the combination of array and intrinsic variables.

Related Topics

Defining Mathematical Functions
Using Valid Operators for Expressions
Using Intrinsic Functions for Expressions
Using Piecewise Linear Functions for Expressions
Using Dataset Expressions

Using Valid Operators for Expressions

The operators that can be used to define an expression or function have a sequence in which they are performed. The following list shows both the valid operators and the sequence in which they are accepted (listed in decreasing precedence):

( )	parenthesis	1
!	not	2
^ (or **)	exponentiation (If you use "**" for exponentiation, as in previous software versions, it is automatically changed to "^".)	3
-	unary minus	4
*****	multiplication	5
/	division	5

+	addition	6
-	subtraction	6
==	equals	7
!=	not equals	7
>	greater than	7
<	less than	7
>=	greater than or equal to	7
<=	less than or equal to	7
&&	logic and	8
||	logic or	8

Related Topics

Defining an Expression

Using Intrinsic Variables

Maxwell recognizes a set of intrinsic variables that can be used to define expressions. Intrinsic variable names are reserved and may not be used as user-defined variable names.

User-defined variables cannot depend on any intrinsic variables.

The following intrinsic variables (shown on the Intrinsic Variables tab after selecting **Project>Project Variables**) may be used to define expressions:

Variable	Units	Description
_Empty		Empty value
_I1 through _I9	mA	Terminal current in an interpretive user-defined model
_t		Variable to define a parametric equation based curve
_u		Variable to define a parametric equation

		based surface
_v		Variable to define a parametric equation based surface
_V1 through _V9	mV	Port voltage in an interpretive user-defined model
Ang	deg	Post-processing variable (cannot be set by the user)
Budget_Index		Post-processing variable (cannot be set by the user)
Distance	mm	
ElectricalDegree	deg	Electrical degree of the rotating machine (cannot be set by the user)
F	GHz	Frequency of the circuit/system analysis
F1	GHz	Frequency of tone 1 in the harmonic balance analysis
F2	GHz	Frequency of tone 2 in the harmonic balance analysis
F3	GHz	Frequency of

		tone 3 in the harmonic balance analysis
FNoi	GHz	Offset noise frequency in the harmonic balance noise analysis
Freq	GHz	Post-processing variable (cannot be set by the user)
Hmax	ns	
Hmin	ns	
Ia and Ib	mA	Post-processing variable (cannot be set by the user)
Index		Post-processing variable (cannot be set by the user)
IWavePhi	deg	Post-processing variable (cannot be set by the user)
IWaveTheta	deg	Post-processing variable (cannot be set by the user)
Normalized Deformation		
Normalized Distance		Post-processing variable (cannot be set by the user)

OP	mW	Post-processing variable (cannot be set by the user)
Pass		Post-processing variable (cannot be set by the user)
Phase	deg	Post-processing variable (cannot be set by the user)
Phi	deg	Post-processing variable (cannot be set by the user)
R	mm	Post-processing variable (cannot be set by the user)
Rho		Post-processing variable (cannot be set by the user)
RSpeed	rpm	Speed of the machine (cannot be set by the user)
Spectrum	GHz	Post-processing variable (cannot be set by the user)
Temp	cel/deg	Analysis temperature (deg)

Tend	ns	
Theta	deg	Post-processing variable (cannot be set by the user)
Time	ns	Time point in a transient analysis
Time0	ns	Time 0 point in a transient analysis
Vac	mV	Post-processing variable (cannot be set by the user)
Vbe	mV	Post-processing variable (cannot be set by the user)
Vce	mV	Post-processing variable (cannot be set by the user)
Vds	mV	Post-processing variable (cannot be set by the user)
Vgs	mV	Post-processing variable (cannot be set by the user)
X, Y, and Z	mm	Post-processing variable (cannot be set by the user)

ZAng and ZRho	deg	Post-processing variable (cannot be set by the user)

Not shown on the Intrinsic Variables tab are the **Speed** and **Position** variables that are available only in Maxwell 2D and 3D transient designs with Motion Setup. These variables are treated as intrinsic variables and are not user-editable. Their values are dynamically calculated during solve time and are expressed in standard SI units. While these variables are *related* to motion setups, they are not necessarily *used* in motion setups. For example, they may also be used with the transient design's excitations. See *Motion Variables Tab* for additional information.

Using Intrinsic Functions in Expressions

Maxwell recognizes a set of intrinsic trigonometric and mathematical functions that can be used to define expressions. Intrinsic function names are reserved and may not be used as variable names.

The following intrinsic functions may be used to define expressions:

Function	Description	Syntax
abs	Absolute value (|x|)	**abs(x)**
sin	Sine	**sin(x)**
cos	Cosine	**cos(x)**
tan	Tangent	**tan(x)**
asin	Arcsine	**asin(x)**
acos	Arccosine	**acos(x)**
atan	Arc tangent. Takes a tangent value as an argument. Because there are two angles in a circle that can have the same tangent value, and atan can return only one value, it returns a value in the range between -90 degrees and +90 degrees (or between -pi/2 and pi/2 in radians).	**atan(x)**
atan2	A two-argument version of the **atan** function. Takes the y and x coordinates (including sign information) of a point as arguments and returns the angle from the X-axis. Can return angle values for the full circle (-180 degrees to +180 degrees or -pi to +pi in radians).	**atan2(y,x)**
asinh	Hyperbolic Arcsine	asinh(x)
atanh	Hyperbolic Arctangent	atanh(x)
sinh	Hyperbolic Sine	**sinh(x)**
cosh	Hyperbolic Cosine	**cosh(x)**
tanh	Hyperbolic Tangent	**tanh(x)**

even	Returns 1 if integer part of the number is even; returns 0 otherwise.	even(x)
odd	Returns 1 if integer part of the number is odd; returns 0 otherwise.	odd(x)
sgn	Sign extraction	sgn(x)
exp	Exponential ($\mathbf{e}^x$)	**exp(x)**
pow	Raise to power (x^y)	pow(x,y)
if	If	**if(cond_exp,true_exp, false_exp)**
pwl	Piecewise Linear with linear extrapolation on x. (pwl can be used with datasets for Design Variables but not for Project variables).	pwl **(dataset_exp, variable)**
pwlx	Piecewise Linear x with linear extrapolation on x	pwlx **(dataset_exp, variable)**
pwl_periodic	Piecewise Linear with periodic extrapolation on x	pwl_periodic **(dataset_exp, variable)**
sqrt	Square Root	**sqrt(x)**
ln	Natural Logarithm (The "log" function has been discontinued. If you use "log(x)" in an expression, the software automatically changes it to "ln(x)".)	**ln(x)**
log10	Logarithm base 10	log10(x)
int	Truncated integer function	int(x)
nint	Nearest integer	nint(x)
max	Maximum value of two parameters	max(x,y)
min	Minimum value of two parameters	min(x,y)
mod	Modulus	mod(x,y)
rem	Returns the fractional part of a decimal number such that rem(x) = x-int(x)	rem(x)
clp	Closest point interpolation. Note: If used with a large 3D dataset, clp function will degrade.	clp (*datasetName*, X,Y,Z)

Note	If you do not specify units, all trigonometric functions interpret their arguments as radians. Likewise, inverse trigonometric functions' return values are in given in radians.

	When the argument to a trigonometric expression is a variable, the units are assumed to be radians. If you want values interpreted in degrees, supply the argument with the unit name **deg**

Related Topics

Defining an Expression

Using Piecewise Linear Functions in Expressions

The following piecewise linear intrinsic functions are accepted in expressions:

```
pwl (dataset_expression, variable)
pwlx (dataset_expression, variable)
pwl_periodic (dataset_expression, variable)
```

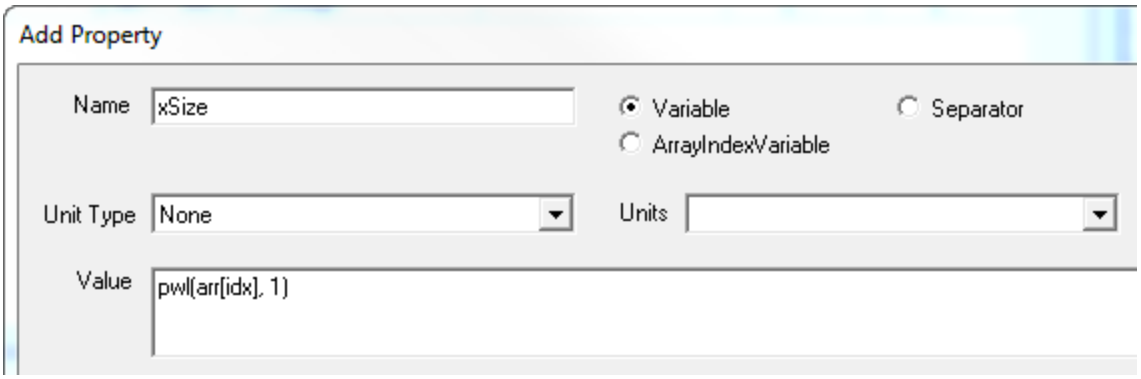
The **pwl** and **pwlx** functions interpolate along the x-axis and returns a corresponding y value. The **pwl_periodic** function also interpolates along the x-axis but periodically.

You can use **pwl** in an expression that uses array variables and datasets for uses such as a frequency dependent material property. (Refer to Adding Datasets for related information on working with datasets.) For example, you specify BulkConductivity as:

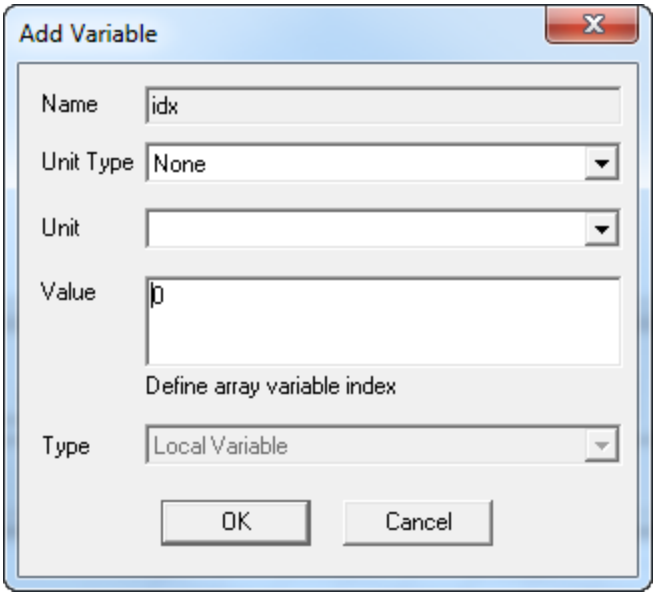
`pwlx($dsArr[$dsIndex], Freq)` where `$dsArr=["$ds1", "$ds2"]`

You can create a Design Variable representing dimension `xSize` as `pwl(arr[idx], 1)` where "`arr`" is an array variable and `idx` is an array index variable.

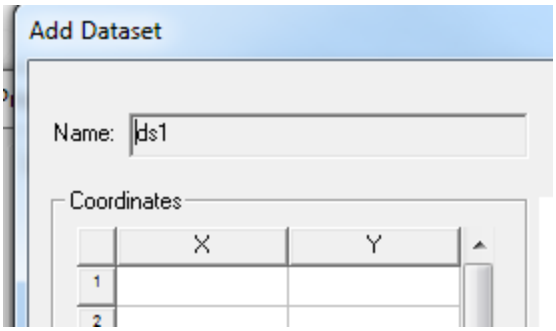
In this case, creating a variable named xSize with pwl(arr[idx], 1) like this:



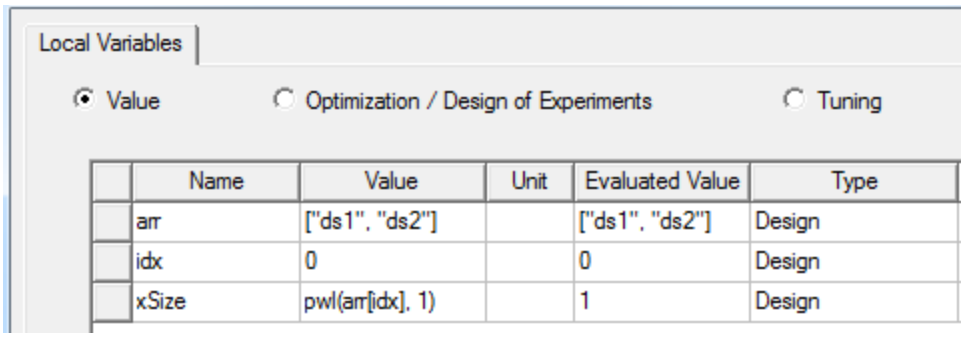
This value for xSize automatically opens a dialog first for the idx index variable:



And then dialogs automatically open for each dataset variable implicit for the predefined array variable:



Then the **Design Properties** dialog shows as follows:



Note	pwl can be used with datasets for Design Variables but not for Project variables.

Related Topics

Adding Datasets
Adding a Design Variable
Modifying Datasets
Using Dataset Expressions

Using Dataset Expressions

Dataset expressions take the following form:

$$\$ds1((x_0,y_0), \ldots, (x_n,y_n))$$

These expressions may be used as the first parameter to piecewise linear (**pwl**, **pwlx** and **pwl_periodic**) functions, and may also be assigned to variables, in which case the variable may be used as the second parameter to **pwl**, **pwlx** and **pwl_periodic** functions.

Dataset expressions are derived from a series of points in a plot created in the **Datasets** dialog box. (Refer to Adding Datasets for related information on working with datasets.) Each plot consists of straight line segments whose vertices represent their end points. A curve is fitted to the segments of the plot, and an expression is derived from the curve that best fits the segmented plot. The created expression is then used in the piecewise linear intrinsic functions.

Related Topics

Adding Datasets
Adding a Design Variable
Modifying Datasets
Using Piecewise Linear Functions in Expressions

Assigning Variables

To assign a variable to a parameter in Maxwell:

- Type the variable name or mathematical expression in place of a parameter value in a **Value** text box.

  If you typed a variable name that has not been defined, the **Add Variable to *DesignName*** dialog box will appear, enabling you to define the design variable.

  If you typed a variable name that included the **$** prefix, but that has not been defined, the **Add Variable** dialog box appears, enabling you to define the project variable.

Note	You can assign a variable to nearly any design parameter assigned a numeric value in Maxwell. See the Maxwell help about the specific parameter you want to vary to determine if can be assigned a variable.

Related Topics

Working with Variables

Choosing a Variable to Optimize

Before a variable can be optimized, you must specify that you intend for it to be used during an optimization analysis in the **Properties** dialog box.

1. If the variable is a design variable, click **Maxwell 2D** (or **Maxwell 3D**, or **RMxprt**)**>Design Properties**. You can also access the Design variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Design Variables**.

   If the variable is a project variable, click **Project>Project Variables**. You can also access the Project variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**.

   The **Properties** dialog box appears.
2. Click the tab that lists the variable you want to optimize.
3. Click the row containing the variable you want to optimize.

Note	Dependent variables cannot be optimized.
4. Select the **Optimization/ Design of Experiments** option.
5. For the variable you want to optimize, select **Include**.

   The selected variable is now available for optimization in an optimetrics setup defined in the current design or project.

Note	Complex numbers are not allowed for variables to be used in an Optimetrics sweep, or for optimization, statistical, sensitivity or tuning setups.

6. Optionally, override the default minimum and maximum values that Optimetrics uses for the variable in every optimization analysis. During optimization, the optimizer does not consider variable values that lie outside of this range.
7. Click **OK**.

Related Topics

Setting up an Optimization Analysis

Including a Variable in a Sensitivity Analysis

Before a variable can be included in a sensitivity analysis, you must specify that you intend for it to be used during a sensitivity analysis in the **Properties** dialog box.

1. If the variable is a design variable, click **Maxwell 2D** (or **Maxwell 3D**, or **RMxprt**)**>Design Properties**. You can also access the Design variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Design Variables**.

   If the variable is a project variable, click **Project**>**Project Variables**. You can also access the Project variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**.

   The **Properties** dialog box appears.
2. Click the tab that lists the variable you want to include in the sensitivity analysis.
3. Click the row containing the variable you want to include in the sensitivity analysis.

Note	Dependent variables cannot be included in a sensitivity analysis.
4. Select the **Sensitivity** option.
5. For the variable you want to include in the sensitivity analysis, select **Include**.

   The selected variable is now available for sensitivity analysis in a sensitivity setup defined in the current design or project.

Note	Complex numbers are not allowed for variables to be used in an Optimetrics sweep, or for optimization, statistical, sensitivity or tuning setups.

6. Optionally, override the default minimum and maximum values that Optimetrics uses for the variable in every sensitivity analysis. During sensitivity analysis, Optimetrics does not consider variable values that lie outside of this range.
7. Optionally, override the default initial displacement value that Optimetrics uses for the variable in every sensitivity analysis. During sensitivity analysis, Optimetrics will not consider a variable value for the first design variation that is greater than this step size away from the starting variable value.
8. Click **OK**.

Related Topics

Setting up a Sensitivity Analysis

Choosing a Variable to Tune

Before a variable can be tuned, you must specify that you intend for it to be tuned in the **Properties** dialog box.

1. If the variable is a design variable, click **Maxwell 2D** (or **Maxwell 3D**, or **RMxprt**)**>Design Properties**. You can also access the Design variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Design Variables**.

   If the variable is a project variable, click **Project**>**Project Variables**. You can also access the Project variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**.

   The **Properties** dialog box appears.
2. Click the tab that lists the variable you want to tune.
3. Click the row containing the variable you want to tune.

Note	Dependent variables cannot be tuned.
4. Select the **Tuning** option.
5. For the variable you want to tune, select **Include**.

Note	Complex numbers are not allowed for variables to be used in an Optimetrics sweep, or for optimization, statistical, sensitivity or tuning setups.
6. Click **OK**.

   The selected variable is now available for tuning in the **Tune** dialog box.

Related Topics

Tuning a Variable

Including a Variable in a Statistical Analysis

Before a variable can be included in a statistical analysis, you must specify that you intend for it to be used during a statistical analysis in the **Properties** dialog box.

1. If the variable is a design variable, click **Maxwell 2D** (or **Maxwell 3D**, or **RMxprt**)**>Design Properties**. You can also access the Design variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Design Variables**

   If the variable is a project variable, click **Project**>**Project Variables**. You can also access the Project variables from a menu in the lower left corner of the following Optimization

dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**.

The **Properties** dialog box appears.

2. Click the tab that lists the variable you want to include in the statistical analysis.
3. Click the row containing the variable you want to include in the statistical analysis.

Note	Dependent variables cannot be included in a statistical analysis.

4. Select the **Statistics** option.
5. For the variable you want to include in the statistical analysis, select **Include**.

   The selected variable is now available for statistical analysis in a statistical setup defined in the current design or project.

Note	Complex numbers are not allowed for variables to be used in an Optimetrics sweep, or for optimization, statistical, sensitivity or tuning setups.

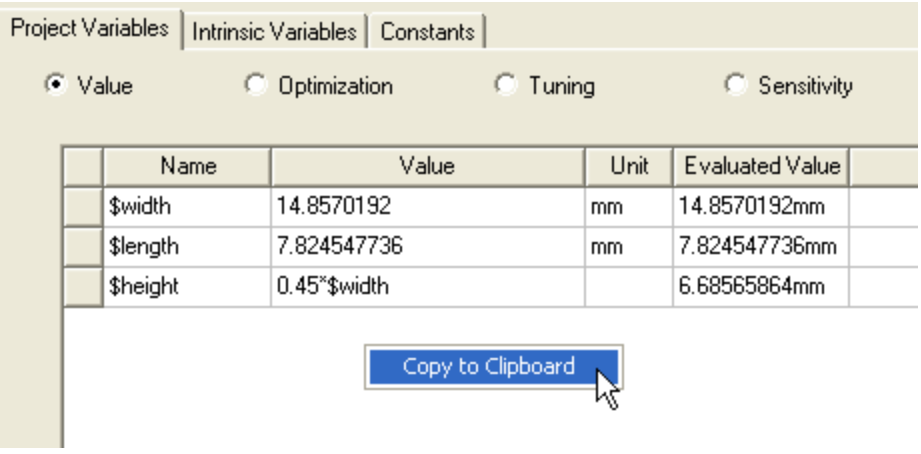
6. Optionally, override the distribution criteria that Optimetrics uses for the variable in every statistical analysis.
7. Click **OK**.

Related Topics

Setting up a Statistical Analysis

Exporting Variables for Documentation

By right-clicking in the Project Variables dialog, you can copy a list of variables and their values to the clipboard. You can then paste these variables elsewhere.



The pasted variables appear in a tab separated column format. Fields that do not contain values are left blank.:

Name	Value	Unit	"Evaluated Value	"Description	Read-only	Hidden
$width	14.8570192	mm	14.8570192mm		false	false
$length	7.824547736	mm	7.824547736mm		false	false
$height	0.45*$width		6.68565864mm		false	false

Related Topics

Assigning Variables

Maxwell ACT Extensions Wizards

The following Maxwell extension Wizards implemented in Ansys ACT can be accessed through the **View>ACT Extensions** user interface.

- Maxwell Eccentricity
- Machine Toolkit

Related Topics

Using the Motion Eccentricity Wizard

Using the Machine Toolkit Wizard

Using the Maxwell Machine Toolkit Wizard (Windows only)

The Machine Toolkit wizard (Windows only) requires a valid 2D or 3D transient design with motion setup. The machine toolkit uses the original design to create a new design which is then used to perform the analysis. The machine toolkit supports five machine types:

- PM (Permanent Magnet) Synchronous Machines
- Induction Machines
- Synchronous Reluctance Machines
- Switched Reluctance Machines
- Wound-rotor Synchronous Machines

The machine toolkit can be used in the following four scenarios. Choose the one appropriate for your design needs.

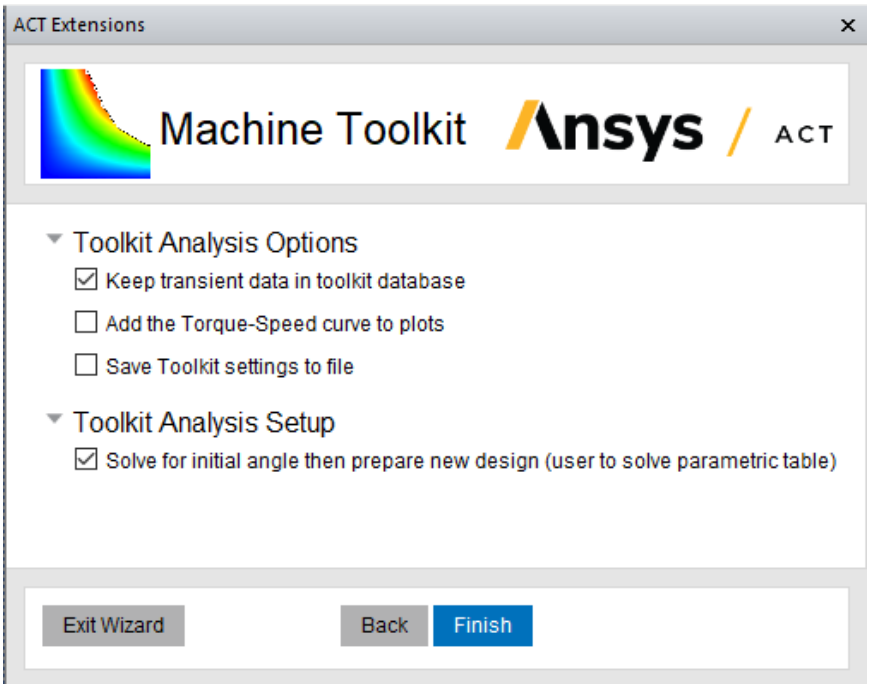
- **Interactive Solve**
    1. For interactive solve, define the machine parameters, then click **Finish**. The toolkit will solve the parametric table sweep, and calculate efficiency maps immediately on

the same machine. Make sure you have set the analysis setting you wish to use (DSO/HPC).

Note	The user does not have to solve the parametric sweep, it is solved automatically.

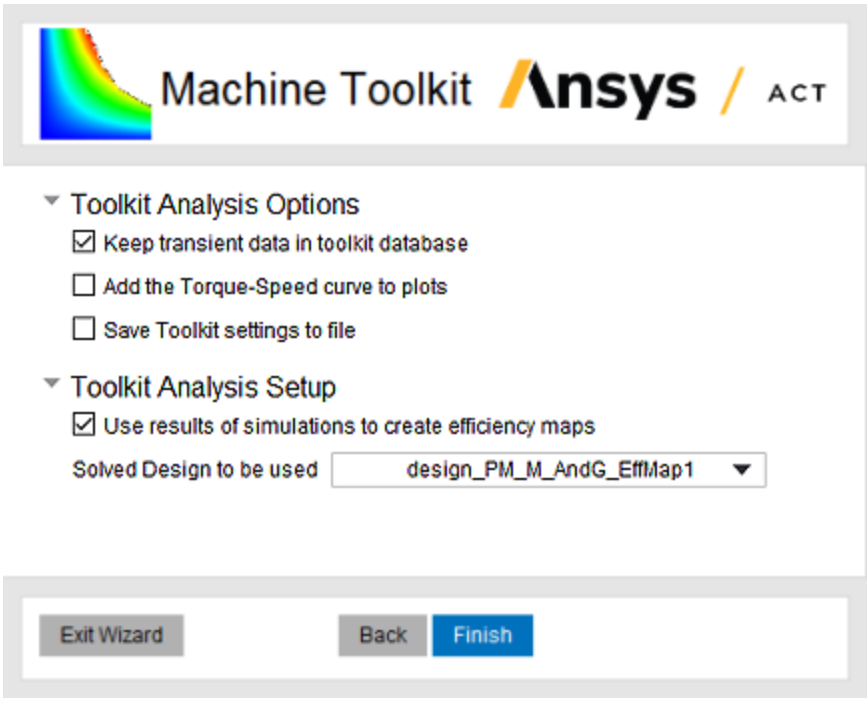
- The toolkit creates a new design.
- Parametric sweep is solved.
- For PM Synchronous machines, Synchronous Reluctance machines, and Wound-rotor Synchronous machines: If not provided by the user, the DQ initial alignment angle is calculated by solving an independent simulation.
- Results are extracted.
- Postprocessing is performed, and reports (contour plots) are created in the design.

- **Delayed Solve**
  1. For delayed solve, define the machine parameters, and check **Solve for initial angle then prepare new design (user to solve parametric table)**.



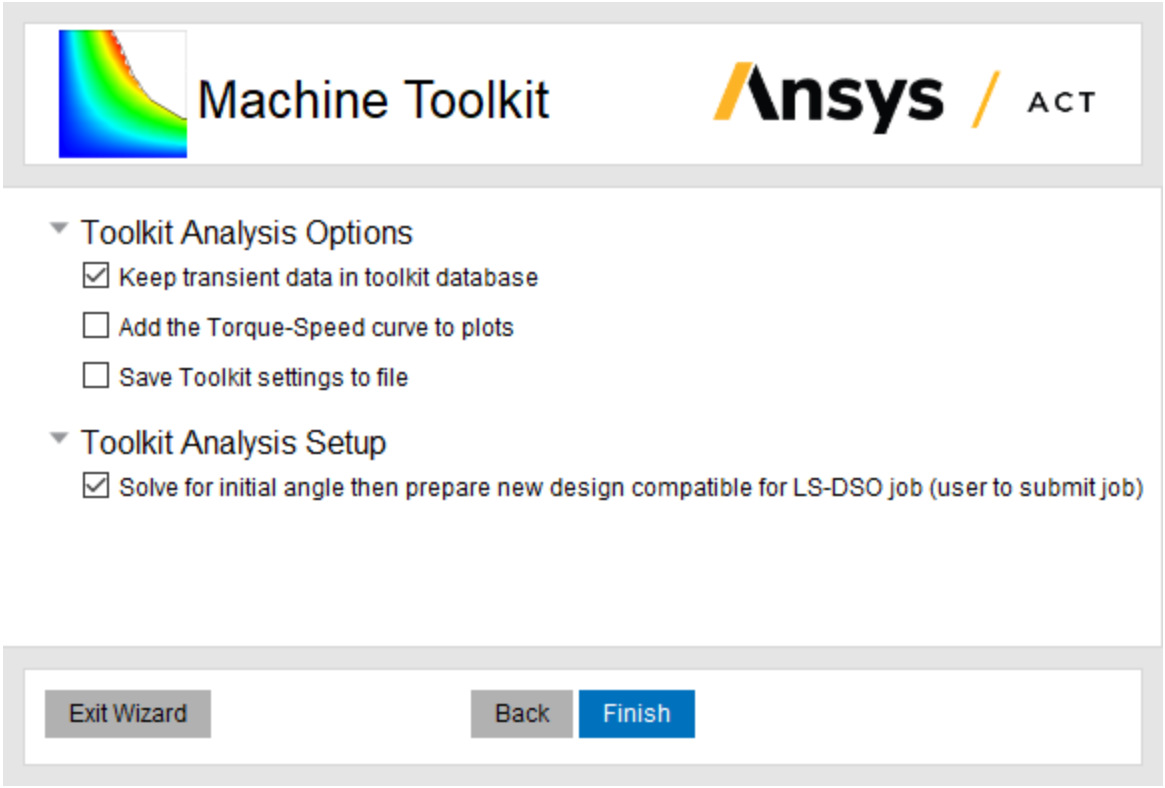
  2. Click **Finish**.
     - A new design is created.
     - For PM Synchronous machines, Synchronous Reluctance machines, and Wound-rotor Synchronous machines: If not provided by the user, the DQ initial alignment angle is calculated by solving an independent simulation.
     - The parametric sweep is then solved separately by the user.
  3. After solving, reopen the toolkit. The parameter entries will be prefilled.

4. Navigate to the last wizard page and select **Use results of simulations to create efficiency maps** and select the **Solved Design to be used**.

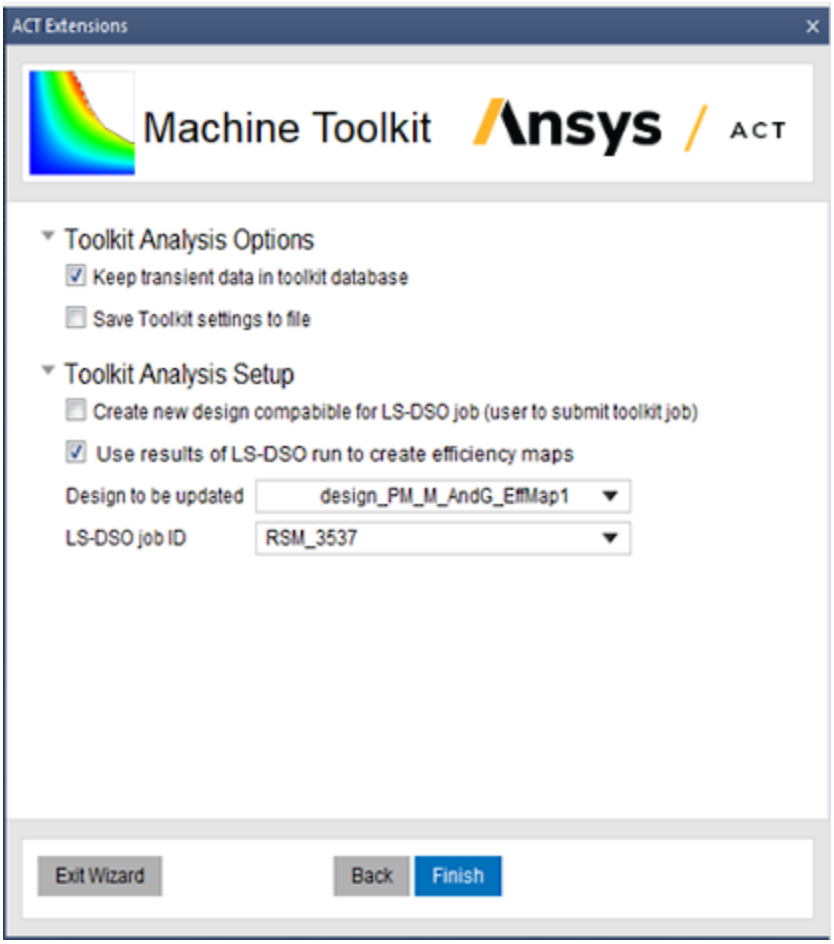


5. Click **Finish**.
    - Results are extracted.
    - Postprocessing is performed, and reports (contour plots) are created in the design.

- **Solve using LS-DSO**
    1. For solve using LS-DSO, define the machine parameters, and check **Solve for initial angle then prepare new design compatible for LS-DSO job (user to submit job)**.



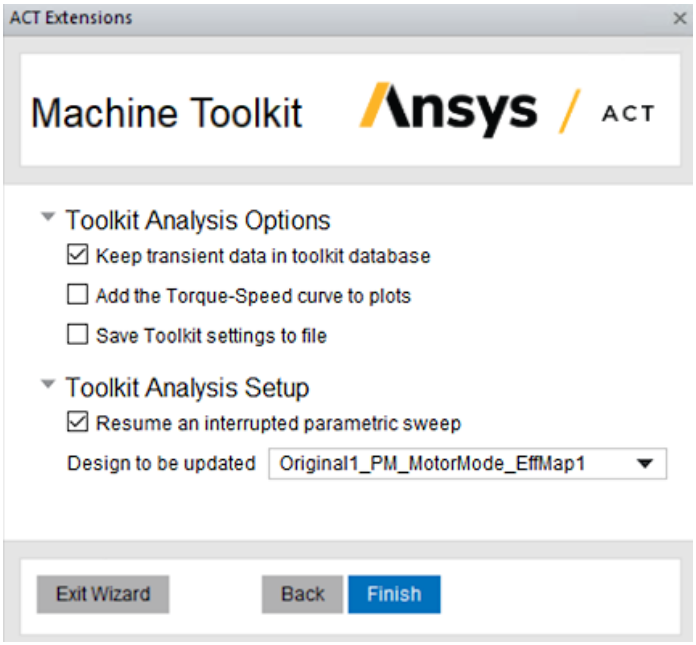
2. Click **Finish**.
    - A new design is created.
    - For PM Synchronous machines, Synchronous Reluctance machines, and Wound-rotor Synchronous machines: If not provided by the user, the DQ initial alignment angle is calculated by solving an independent simulation.
    - The new design contains a parametric sweep that needs to be solved.
    - The toolkit stops to let user solve the LS-DSO job separately.
3. Upon completion of the LS-DSO job, restart the toolkit. The parameter entries will be prefilled.
4. Navigate to the last wizard page and select **Use results of LS-DSO run to create efficiency maps**, then select the **Design to be updated** and also the **LS-DSO job ID** that was used for the job.



5. Click **Finish**.
    - Results are extracted.
    - Post-processing is performed, and reports (contour plots) are created in the design.

- **Resume an Interrupted Parametric Sweep Run**
    1. For this scenario, first select an existing original design. The parameters of the interrupted efficiency map design are automatically prefilled in the UI.
    2. Navigate to the last wizard page and select **Resume an interrupted parametric sweep** and select the **Design to be updated**.
    3. The parametric sweep of this design will start from where it was interrupted, and

efficiency maps will be generated in post processing.



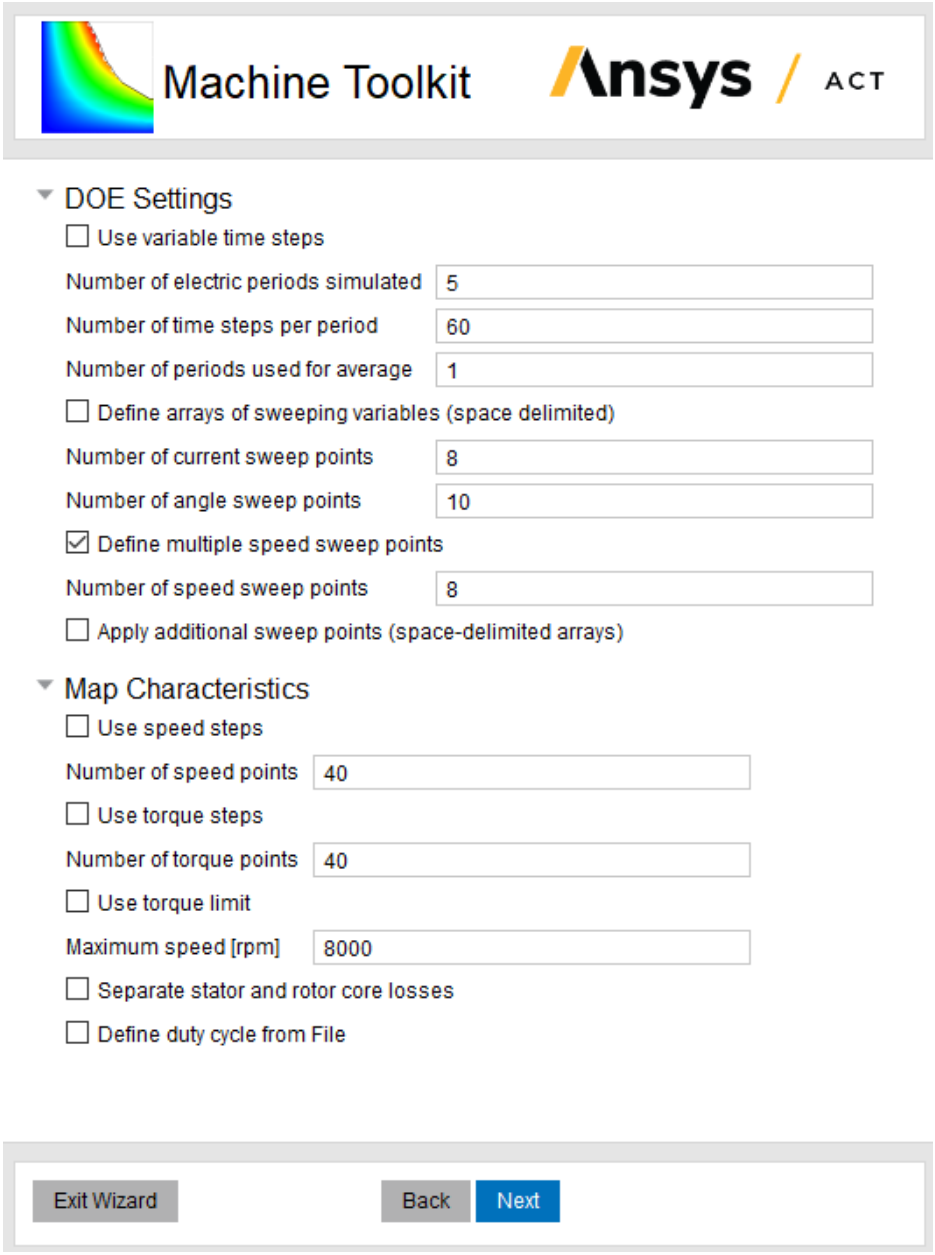
- **Generate Efficiency map only**
  1. For this scenario, first select an existing design. The parameter entries will be prefilled.
  2. Navigate to the last wizard page and select **Use results of simulations to create efficiency maps** and select the **Solved Design to be used**.
  3. Both general transient and periodic (or half-periodic) TDM can be used to conduct parametric analysis and create performance maps. Define the TDM option in the analysis setup of the original design. When conventional transient method or general transient TDM is used to extract performance data, it is necessary to define Number of electric periods simulated and Number of periods used for average in DOE settings. There is an option for the user to define two time intervals, and also specify the number of electric periods and number of time steps per period for each interval. When periodic or half-periodic TDM is enabled in the original design, such options are hidden in DOE settings, and the Machine Toolkit automatically defines the time duration for each simulation.

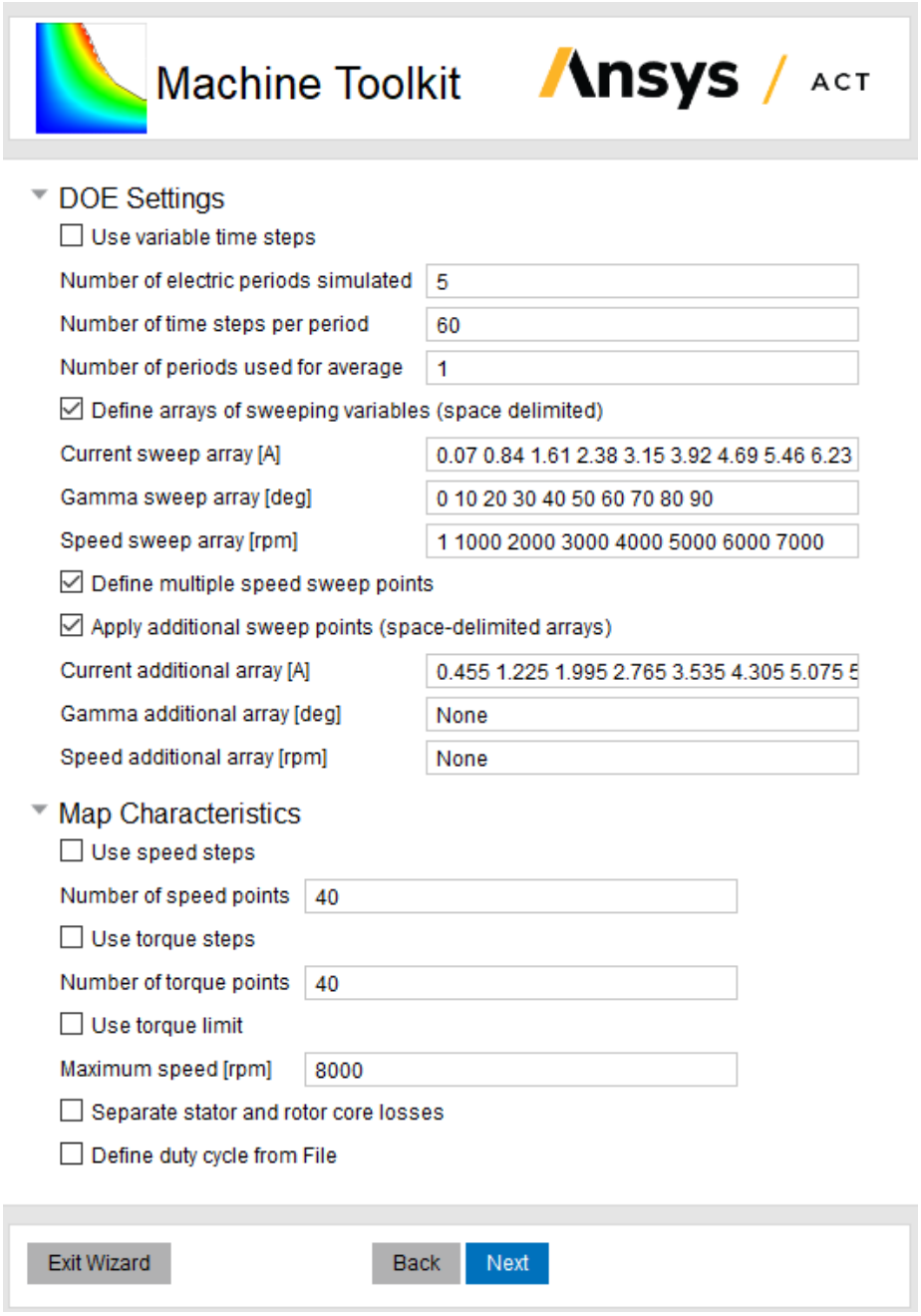
     Machine Toolkit provides two options for the user to determine sweep points in the parametric analysis. The first option is to define the number of sweep points for each independent variable, and then Machine Toolkit assigns the sweep points that are uniformly distributed in the solution domain. The second option is to define the numerical arrays of the sweeping variables, where each value is separated from its adjacent values by a single space character.

     Once the parametric analysis of the prior efficiency map simulation is completed, there is an option for the user to apply additional sweep points to the parametric

analysis. The operating points that are already computed in the prior parametric analysis are automatically skipped, and Machine Toolkit only simulates these additional points.

Note	• For an induction machine model with general transient analysis, the simulations of some operating points in the prior parametric sweep may not be skipped. • TDM options cannot be enabled for Switched Reluctance machines.





For induction machines, to use periodic TDM or half-periodic TDM, all sources including induced eddy current have to be in the same frequency. This means periodic TDM or half-periodic TDM is only applicable when rotor is locked at zero speed. To this end, we can introduce equivalent conductivity σ_eq associated with locked rotor to approximate the performance at original speed (σ_eq = σ * slip). For this kind of equivalence, fundamental component will be correctly modelled. This approximation is normally good enough to the simulated performances including

torque, current, voltage, etc with one exception that computed loss in the rotor needs to be rescaled in terms of slip.

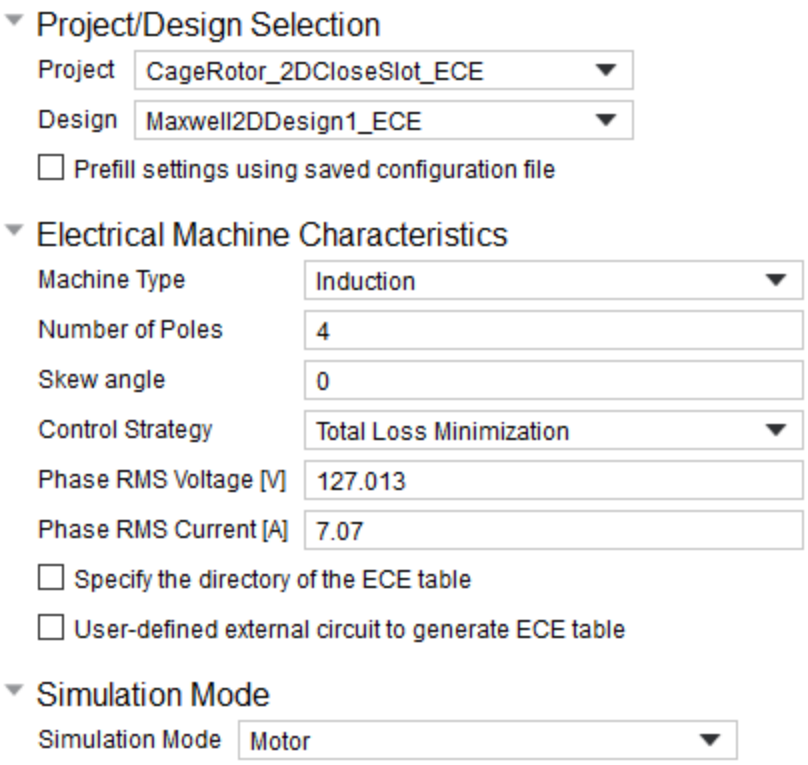
Note	The following steps must be done before proceeding to use periodic TDM solver in the toolkit: a. Add a project parameter called **$slip**. The **Value** and **Evaluated Value** can be any number since the parametric sweep will override it. b. Modify the rotor bar's material, and change the conductivity from **X** to **X*$slip**. c. Modify the end connection resistance from **X** to **X/$slip**. d. Select **Enable two level**for HPC setting.

4. Click **Finish**.
   - Results are extracted.
   - Post-processing is performed, and reports (contour plots) are created in the design.

- **Create Efficiency Maps based on Reduced-Order Model**

  Machine Toolkit supports efficiency map generation based on the reduced-order model (ROM) for induction machines. This feature must be implemented in a design with Eddy Current solution type. The following parameters must be defined through the Machine Toolkit UI about the characteristics of the efficiency maps, including phase RMS voltage, maximum torque and speed, core loss correction factor, control strategy, simulation mode, friction loss, windage loss and their reference speed.

  Refer to the item ECEIM_Model for details about its input parameters and the mathematical representations of the ECE ROM for induction machines. All identified parameters of the ECE ROM are saved in the file `ece_table.txt`, and the Machine Toolkit needs this file to generate efficiency maps. There are three options for the user to obtain this file and define its directory in the Machine Toolkit. The first option is that the user directly specifies the file by checking **Specify the directory of the ECE table**. The second option is that the user first define an external circuit which includes an ECEIM_Model module, then the Machine Toolkit launches the analysis and automatically obtains the file `ece_table.txt`. The third option is that the Machine Toolkit creates an external circuit with the ECEIM_Model module and imports it to the definition of Excitations, then runs the simulation to get this table. Additional parameters must be input by the user if the third option is selected, including the skew angle, phase RMS current, number of current sweep points, and the resistance and end inductance of the windings. The user can also define the current sweep array for the ECEIM_Model.



Related Topics

Defining Machine Toolkit Parameters
Working with the ACT Extensions Window
Permanent-Magnet Synchronous Machines
Three-Phase Induction Machines

Defining Machine Toolkit Parameters

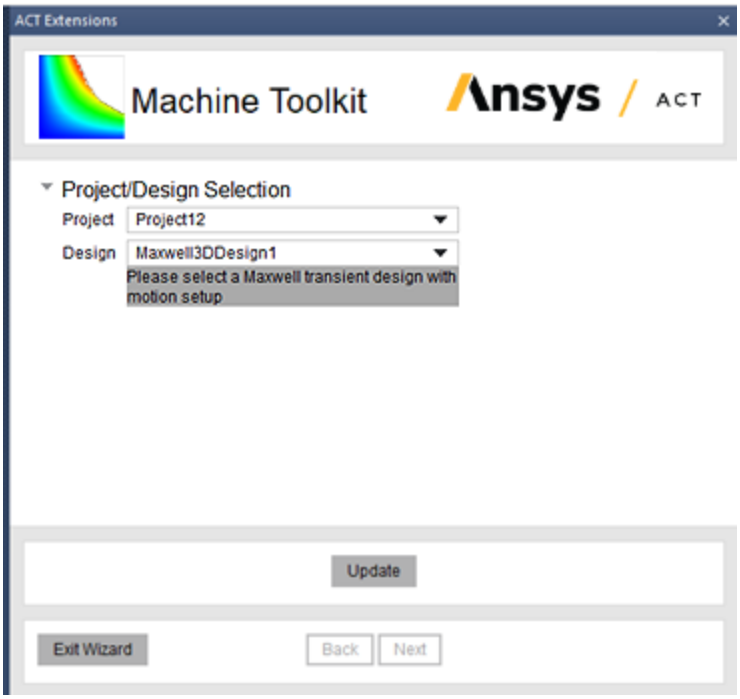
The machine toolkit supports five machine types:

- PM (Permanent Magnet) Synchronous Machines
- Induction Machines
- Synchronous Reluctance Machines
- Switched Reluctance Machines
- Wound-rotor Synchronous Machines

Note	The Machine Toolkit requires a valid 2D or 3D transient design with motion setup. The machine toolkit uses the original design to create a new design which is then used to perform the analysis.

The general workflow for using the Machine Toolkit to define machine parameters is as follows:

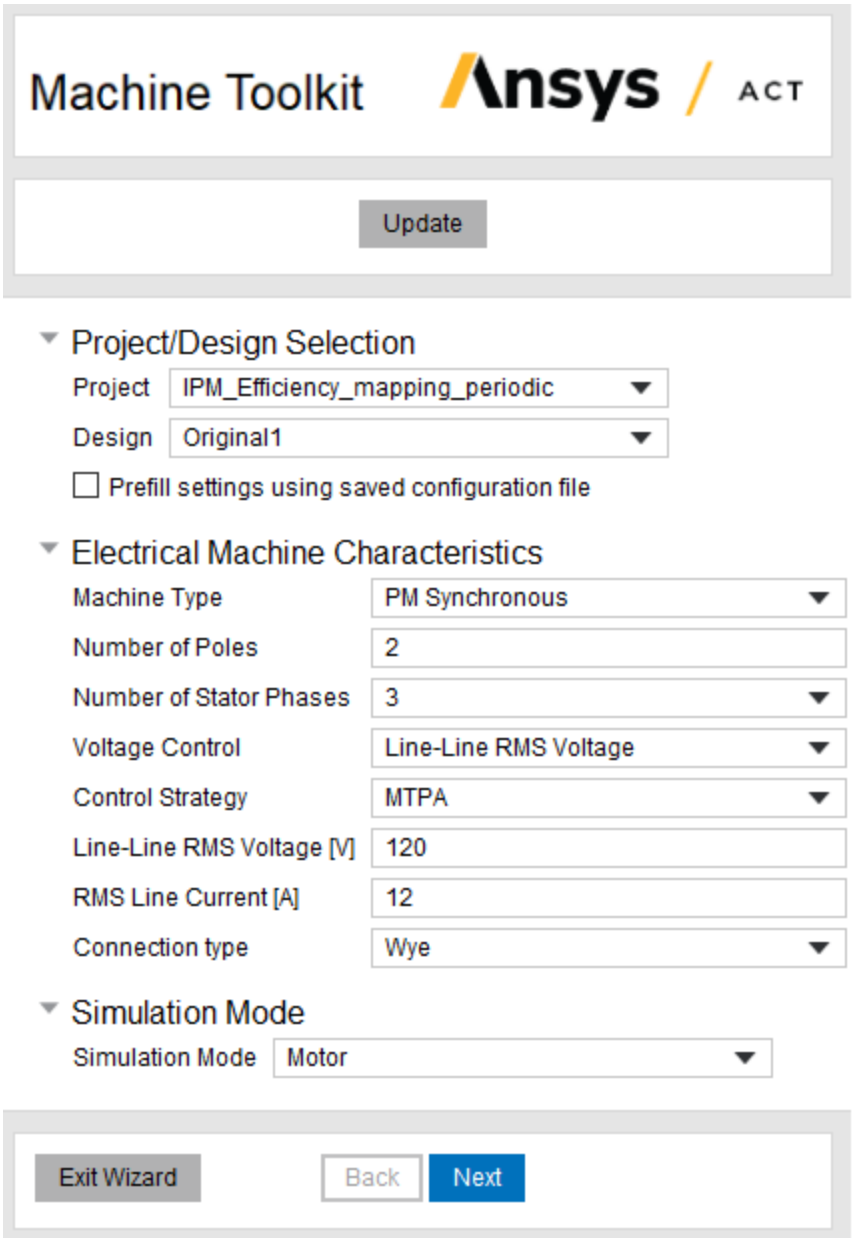
1. In the ACT Extensions Wizards window, click on the Machine Toolkit icon to launch the wizard, and select a valid transient design with motion.
2. You can open a project/design in the project manager if one is not already open.



After loading the project, select **Update** to refresh the Project/Design selections.

3. Select the desired Project and Design. Only designs that do not use reserved local variables are shown in the drop-down list. Reserved local variables are:

   Speed_TSC, Imax_TSC, Gamma_TSC, Vmax_TSC, Slip_TSC, Freq_TSC, and Field_TSC.



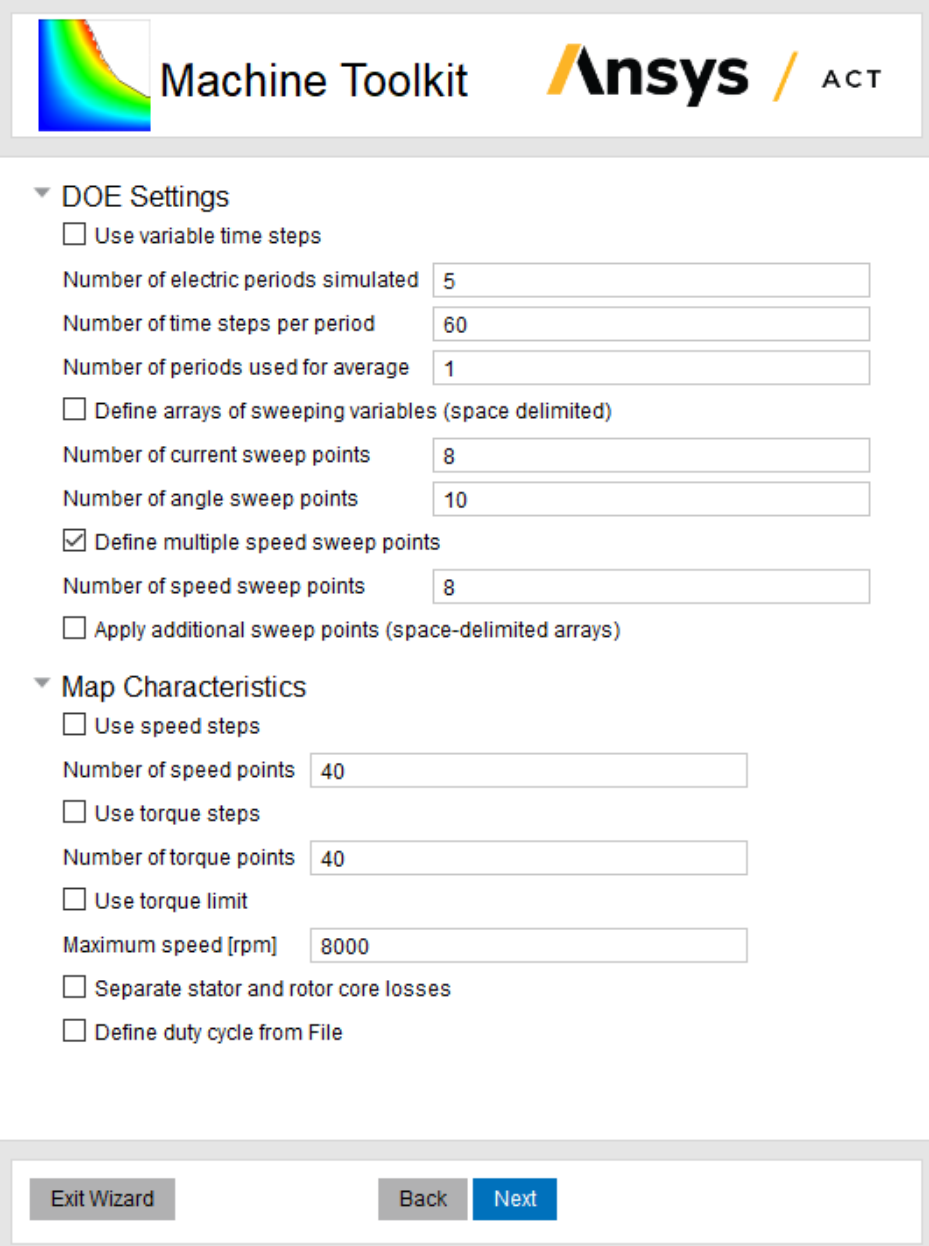
4. Enter all characteristics of the machine: **Machine Type** (**PM Synchronous**, **Induction**, **Synchronous Reluctance**, **Switched Reluctance**, or **Wound-rotor Synchronous**), **Number of Poles**, etc., in the various fields. Also select the desired **Simulation Mode** (**Motor** or **Generator**).

Note	• You can pre-fill all settings using a previously saved configuration `.sav` file by selecting **Prefill settings using saved configuration file**. • If you have previously used the toolkit with the current project, the toolkit prefills entries automatically.

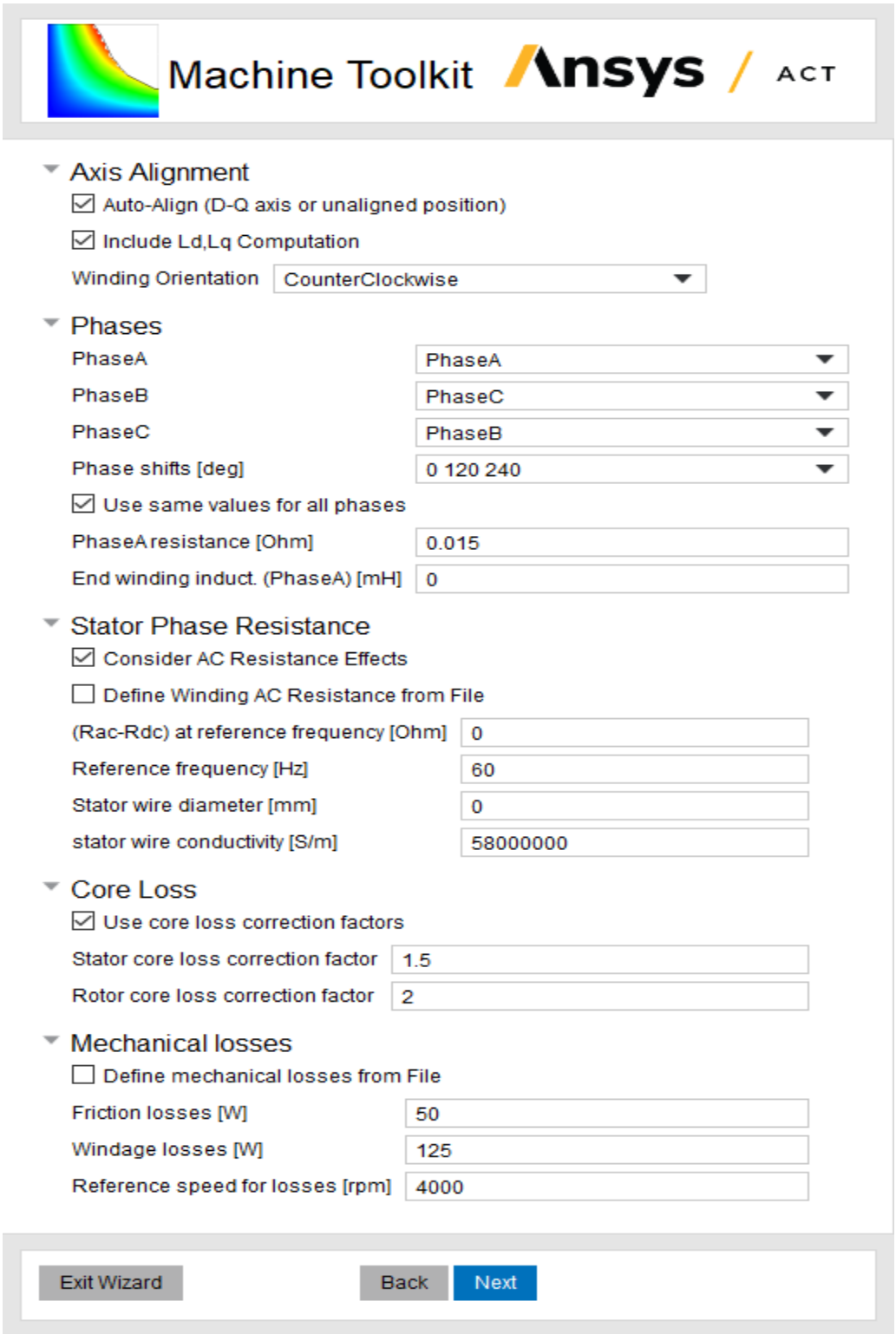
5. Click **Next** to go to the next page of the wizard. Here you enter the desired **DOE** (design of experiments) **Settings**, and the **Map Characteristics** used to generate the efficiency map when the design is solved.

Note	The content of the DOE Settings and Map Characteristics change based on which machine type you selected on the previous page.

Note that check boxes allow you to **Use speed steps** and **Use torque steps** instead of using the default speed and torque points. A parametric sweep will be created to account for the number of points (or steps).

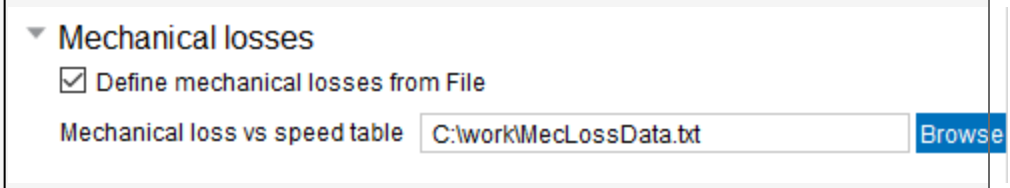


6. When finished, click **Next** to move to the next page of the wizard.



- For PM Synchronous machines, Synchronous Reluctance machines, and Wound-rotor Synchronous machines, you can provide the alignment angle or have the toolkit calculate it. No matter which simulation mode is chosen, the initial angle must be determined first.
- For Phases, provide phase names, phase shifts, and other phase information.
- Optionally, provide custom loss information (Core Loss, Stator Phase Resistance winding losses, and Mechanical Losses).

Note	There are three options for AC winding loss calculation: • The first option is to apply the analytical AC winding loss model, and the user is required to input the stator wire diameter and conductivity, as well as AC resistance at the reference frequency. • The second option is to use the Litz wire model. • The third option is that the user can input a .txt file that provides a frequency vs. AC resistance table; in this file, the first line should include the names and units of frequency and resistance, e.g., "Freq [Hz] Rac[ohm]", followed by combinations of frequency vs. resistance data in each line, e.g., "60 1.5".

Note	Mechanical losses include friction and windage losses at a reference speed. You can optionally "Define mechanical losses from File", then specify the location of a user-defined “Mechanical loss vs speed table” text file. 

7. When finished, click **Next** to move to the next page of the wizard.
   - It is recommended to have the **Keep transient data in toolkit database** option enabled to retain the data because the extraction process can be lengthy.
   - Check **Save Toolkit settings to file** if you want to save all settings to a `.sav` file for use with another project. (For the current project, the settings are saved automatically.)
   - You can click **Back** to review or edit settings on previous pages.
   - If you choose to **Exit Wizard**, all settings are automatically saved and loaded the next time the current project is loaded.
   - **Toolkit Analysis Setup** options are used for the various scenarios described in *Using the Maxwell Machine Toolkit Wizard*.

8. Click **Finish** to complete the process for the various scenarios as described in *Using the Maxwell Machine Toolkit Wizard*.

Related Topics

Using the Maxwell Machine Toolkit Wizard

Working with the ACT Extensions Window

Permanent-Magnet Synchronous Machines

Three-Phase Induction Machines

Using the Maxwell Eccentricity Wizard

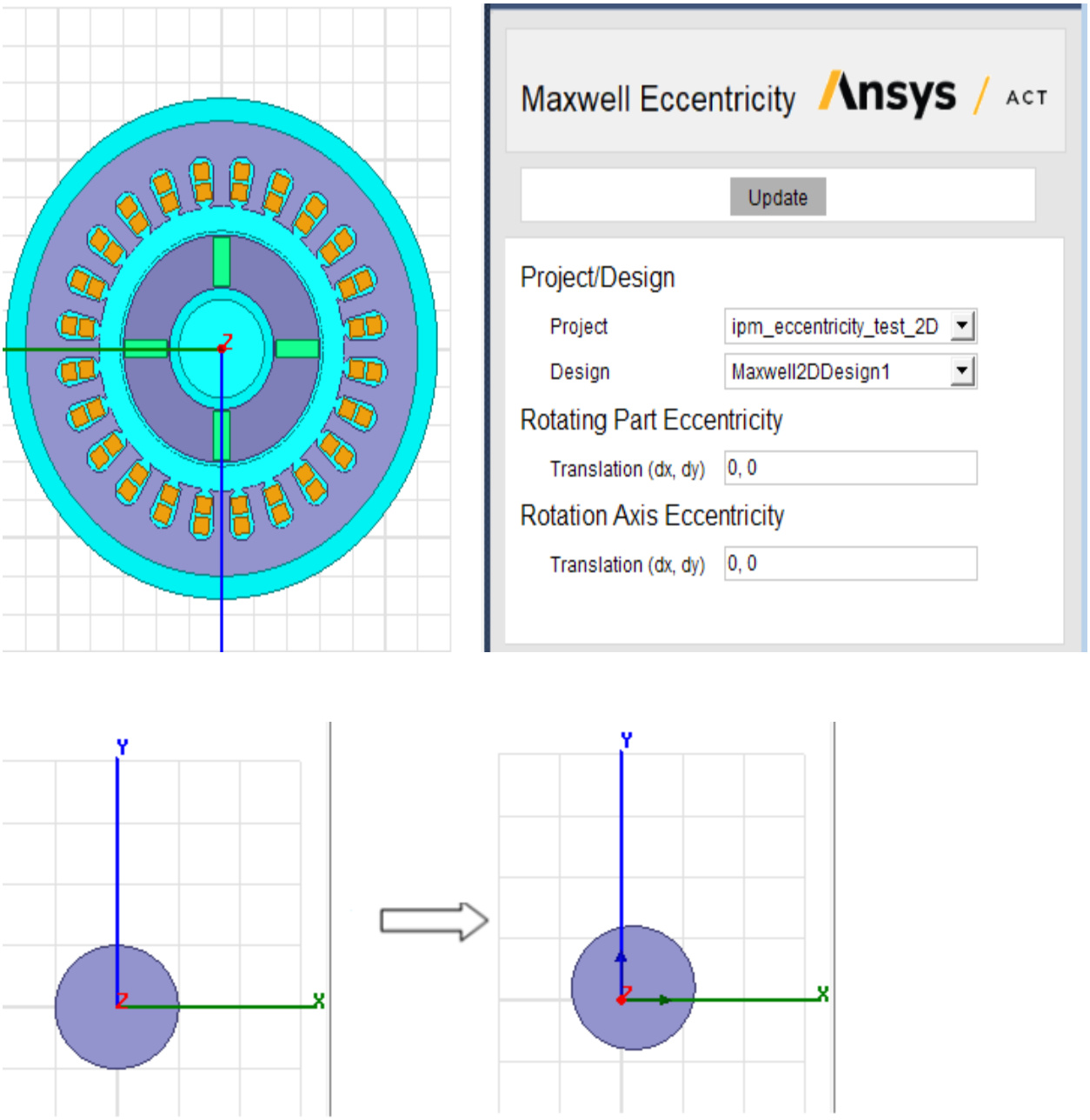
You can use the Maxwell Eccentricity wizard to model both 2D and 3D transient designs that have cylindrical rotational motion set up.

Note	If you want to generate a 3D rotational eccentricity motor design from an RMxprt-generated Maxwell 3D design, you must enter **Eccentricity 1** in the RMxprt **Design Setting** dialog **User Defined Data tab** for the RMxprt design *before* generating the Maxwell 3D design.

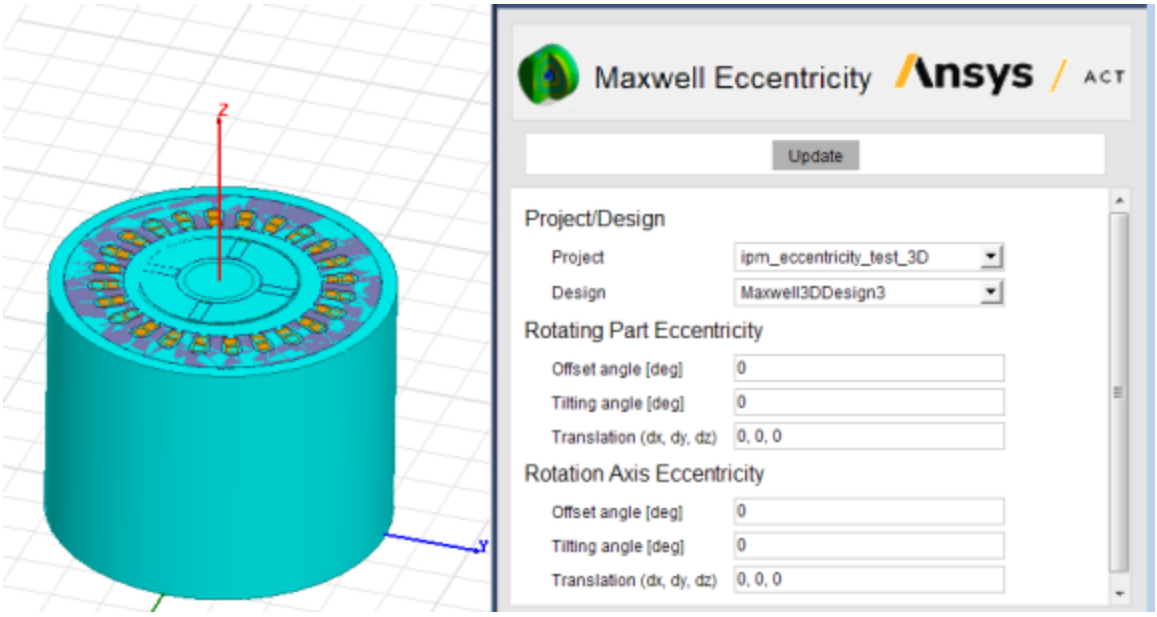
1. Select **View** > **ACT Extensions** to open the ACT Extensions window.
2. Click the **Wizards** button to open the Wizards page, then click **Maxwell Eccentricity** to load the Maxwell Eccentricity setup page.
3. Select the **Project** and **Design** you want to use.

Note	• The **Update** button synchronizes the Project and Design selection lists with those currently loaded in the Desktop. • You can click **Exit Wizard** to close the eccentricity wizard and return to the wizard start page without executing any changes.

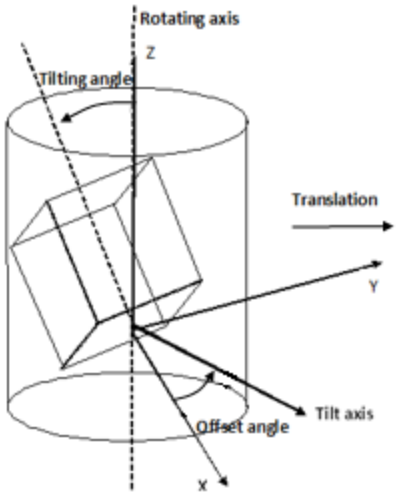
4. Depending on whether your design is 2D or 3D, do one of the following:
    - For a 2D Maxwell project, enter the desired dx and dy translation values to set up the Rotating Part Eccentricity and Rotation Axis Eccentricity. The unit of measure for translation values is the model's unit, which is set using **Modeler** > **Units**.



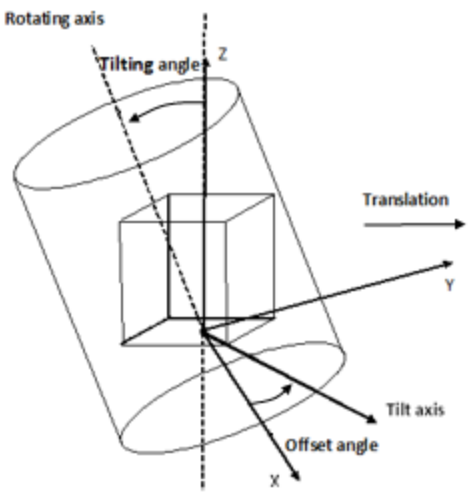
- For a 3D Maxwell project, enter the desired dx, dy , and dz translation values, the offset angle, and the tilting angle for the rotating part eccentricity axis and rotation axis. The unit of measure for translation values is the model's unit, which is set using **Modeler > Units**.



Rotation Part Eccentricity



Rotation Axis Eccentricity



5. Click the **Finish** button at the bottom of the page. The page closes and the wizard adds a new design in the same project, in which the Maxwell geometry has been updated in accordance with the eccentricity setup values you entered as follows:

For 2D designs

- Design variables are created for each of the data entered in the wizard.

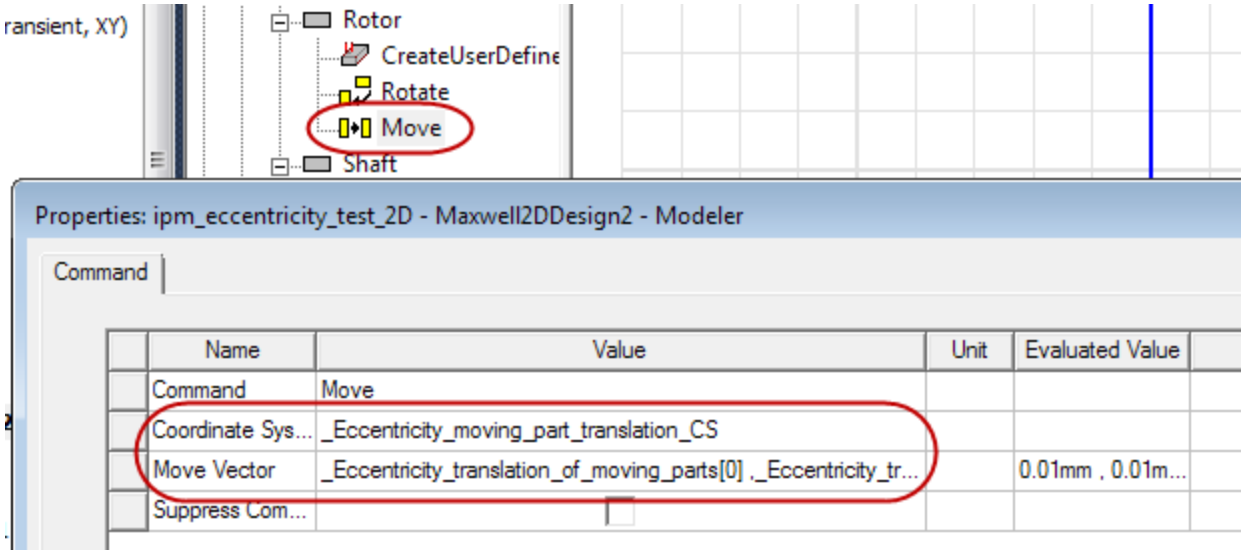
  For rotating part eccentricty:

_Eccentricity_translation_of_moving_parts

For rotation axis eccentricity:

_Eccentricity_translation_of_rotating_axis

- The moving parts in the design motion setup are shifted according to the data defined for **Rotating Part Eccentricity** in the Wizard. New coordinate systems are generated as the object coordinate systems for moving parts, according to the same operations.
- The band is translated according to the data defined for **Rotation Axis Eccentricity** in the wizard.
- The **Moving Vector** entry in the design's motion setup is replaced with the new direction (represented by an axis of a new coordinate system) generated according to the data defined for **Rotation Axis Eccentricity** in the Wizard.



- New design coordinate systems are created.

  _Eccentricity_rotating_axis_translation_CS

  _Eccentricity_moving_part_translation_CS

  _Eccentricity_moving_part_CS_*<part name>* – one for each moving part

Note	• All the above operations of translation, as well as the creation of new coordinate systems, are dependent on the eccentricity design variables. • You can modify these design variables to adjust the eccentricity. • The newly added coordinate systems should not be changed manually by the user. • Using the wizard to modify a design that has already been set up for eccentricity is not recommended.

For 3D designs

- Design variables are created for each of the data entered in the wizard.

  For rotating part eccentricty:

  _Eccentricity_offset_angle_of_moving_parts

  _Eccentricity_translation_of_moving_parts

  _Eccentricity_tilting_angle_of_moving_parts

  For rotation axis eccentricity:

  _Eccentricity_offset_angle_of_motion_axis

  _Eccentricity_translation_of_motion_axis

  _Eccentricity_tilting_angle_of_motion_axis
- The moving parts in the design motion setup are rotated and shifted according to the data defined for **Rotating Part Eccentricity** in the Wizard. New coordinate systems are generated as the object coordinate systems for moving parts, according to the same operations.
- The band is rotated and translated according to the data defined for **Rotation Axis Eccentricity** in the wizard.
- The **Moving Vector** entry in the design's motion setup is replaced with the new direction (represented by an axis of a new coordinate system) generated according to the data defined for **Rotation Axis Eccentricity** in the Wizard.
- New design coordinate systems are created.

  For the motion moving vector:

  _Eccentricity_motion_axis_CS

  _Eccentricity_motion_axis_tilting_axis_CS

  For the moving parts:

  _Eccentricity_moving_part_tilting_axis_CS

  _Eccentricity_moving_part_translation_CS

  _Eccentricity_moving_part_CS_*<part name>* – one for each moving part

Note	• All the above operations of rotation and translation, as well as the creation of new coordinate systems, are dependent on the eccentricity design variables. • You can modify these design variables to adjust the eccentricity. • The newly added coordinate systems should not be changed manually by the user. • Using the wizard to modify a design that has already been set up for eccentricity is not recommended.

Note	When modeling motion eccentricity, the position and direction of the moving objects and/or band are adjusted based on the eccentricity setup. In such cases, all the adjustments are referenced to the original position and direction of the geometry of each

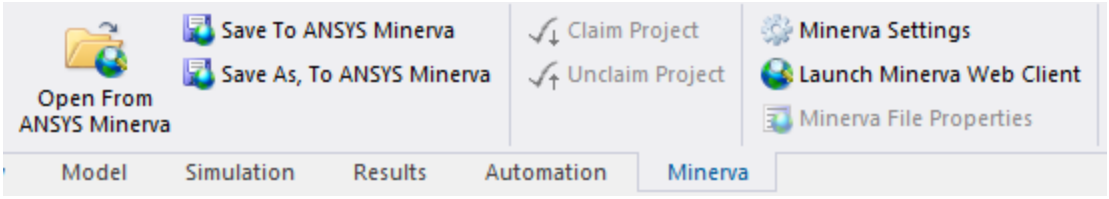
	object, As a result, after you finish the eccentricity setup and click the **Finish** button, all the geometries in the design created by the wizard are updated to new positions based on the eccentricity setup. If you want to edit geometry, you should do so in the original design. After editing the geometry, you can then re-do the eccentricity setup to create a new design.

Related Topics

Modeling Motion Eccentricity
Working with the ACT Extensions Window

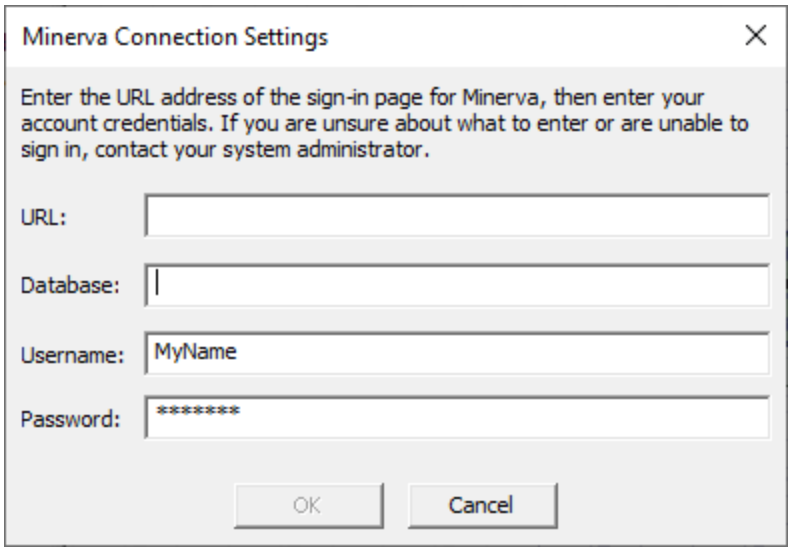
5 - Minerva Remote Storage Environment

Ansys Minerva is a remote storage environment where you can store Ansys project archive files. You can collaborate with other users by downloading a project, making changes, and then uploading the project. One of the features of Ansys Minerva is the ability to store and retrieve Ansys projects from the Minerva server. This ability has been integrated into Electronics Desktop so it is now possible to download and open a project stored on Minerva directly from the Electronics Desktop interface. In the same way, it is also possible to save a project to Minerva. The image below shows the Minerva ribbon containing Minerva commands accessible through the desktop.



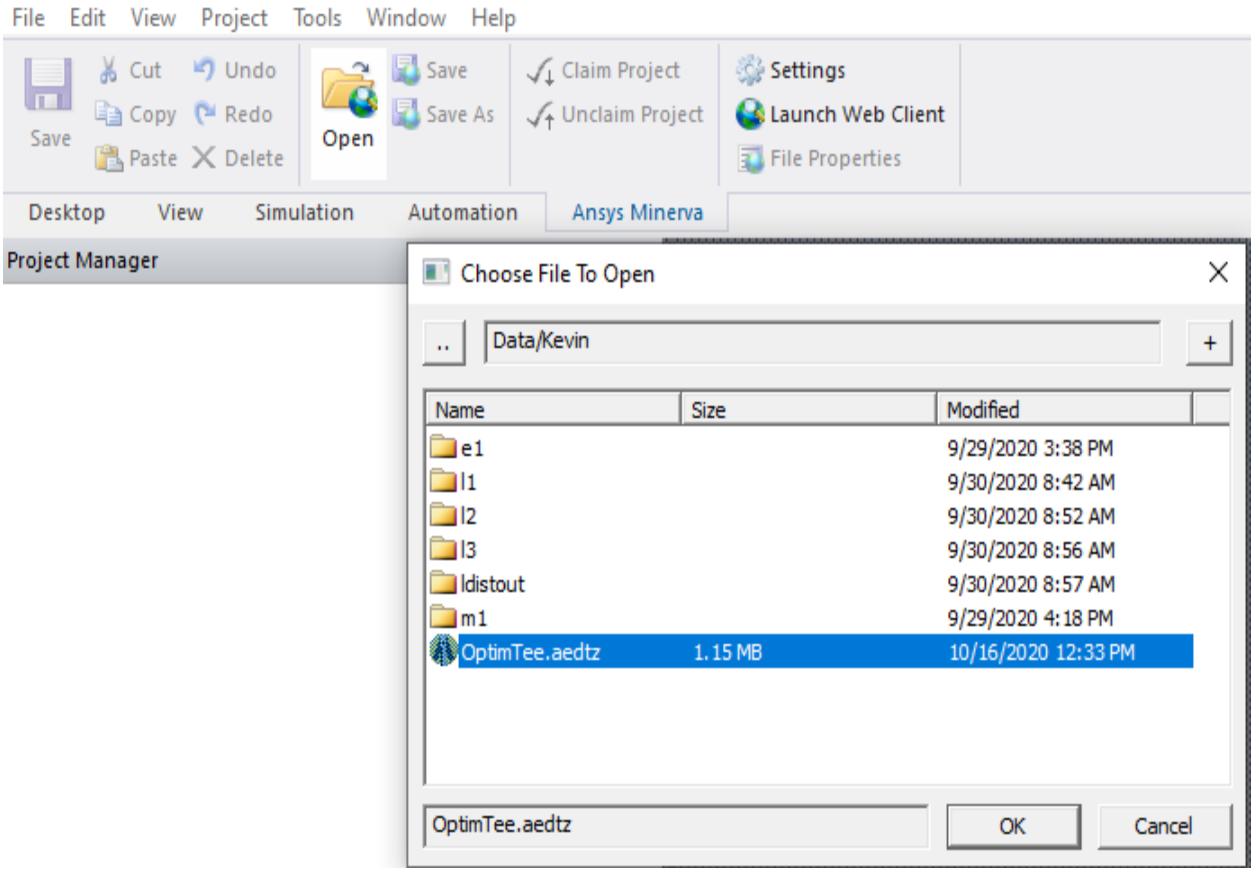
Minerva has its own HTML web interface with additional features as a knowledge management application that secures critical simulation data, and provides simulation process and decision support to simulation teams across geographies and functional silos.

Minerva Settings

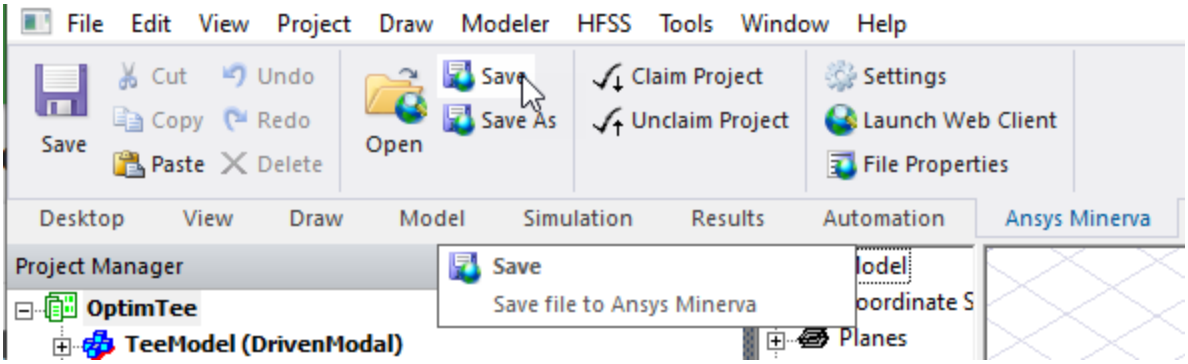


When you first login via the desktop, or to the Web Client, you must provide a Minerva URL, a database name, a user name, and a password. You will need to obtain these from your system administrator. Once you provide the connection settings, the other Minvera commands are enabled.

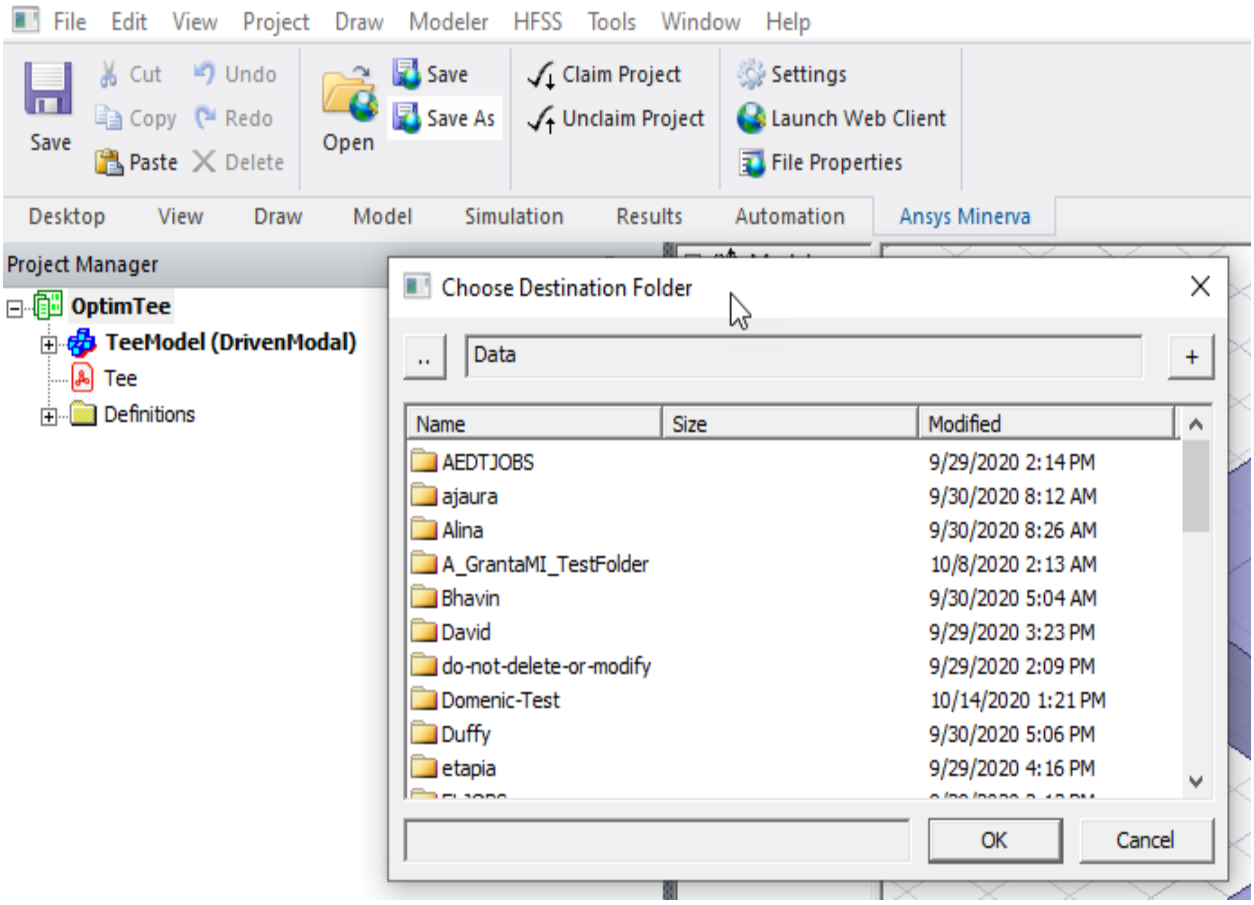
Open From Ansys Minerva: opens the Choose File to Open browser that lets you select and download an archive file, extract it, and open it. The [..] button moves up one level in the file hierarchy. The [+] button creates a new folder at the current level.



Save: archives the current project, and saves it to a remote Minerva system.



Save As...: prompt for remote file, archive project, save to remote Minerva system.



In order to prevent users from overwriting each other's work, Minerva has added the concept of claiming and unclaiming files. A file claimed by one user cannot be overwritten on the Minerva server by another user. Unclaimed files can be overwritten by anyone at any time. Once a file has been claimed by one user, it cannot be claimed by a different user until it has been unclaimed by the original claimer. Note: even if a project is claimed by one user, other users are free to download, open and edit the file locally.

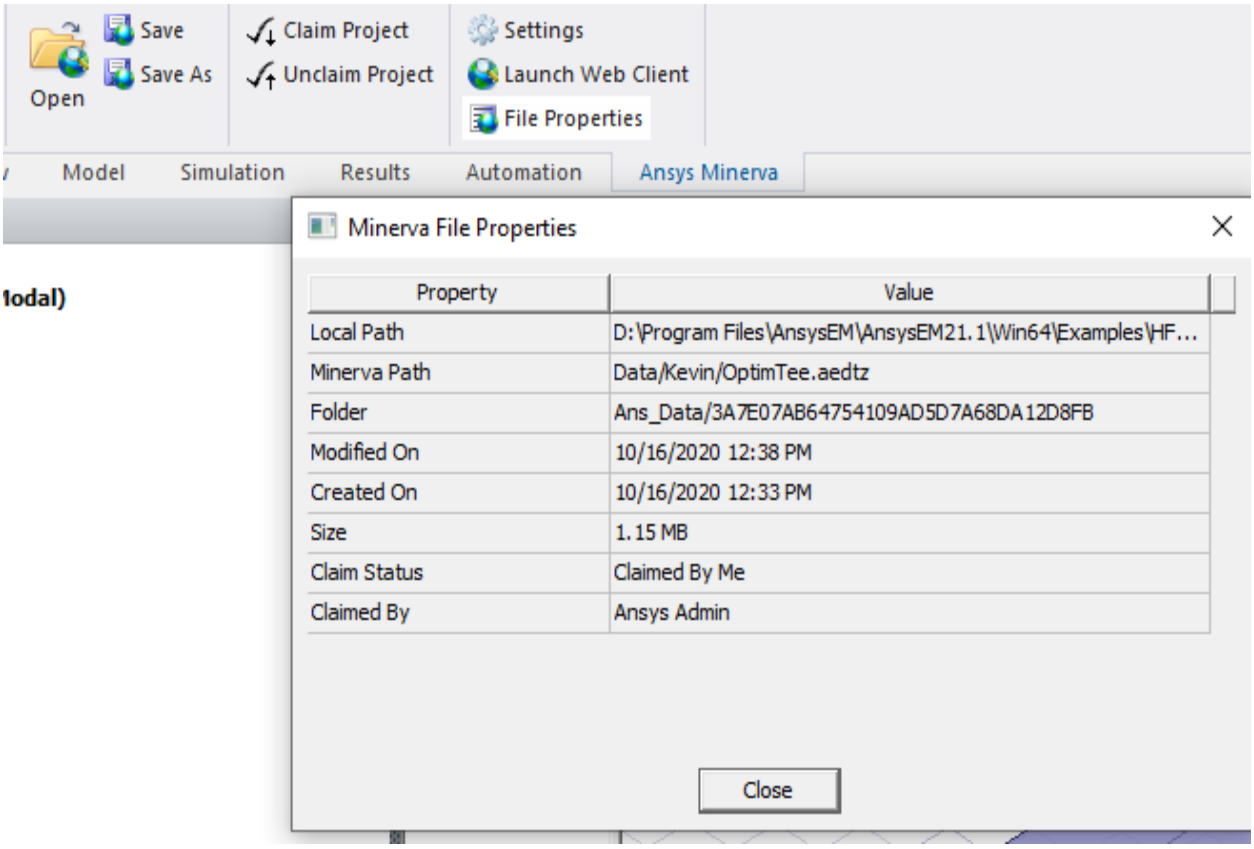
The recommended flow when working on a project is:

1. Claim the file, making sure you are working with the most recent version.
2. Make edits.
3. Save the file to Minerva.
4. Unclaim the file.

Claim Project: claims file so it cannot be modified by other users.

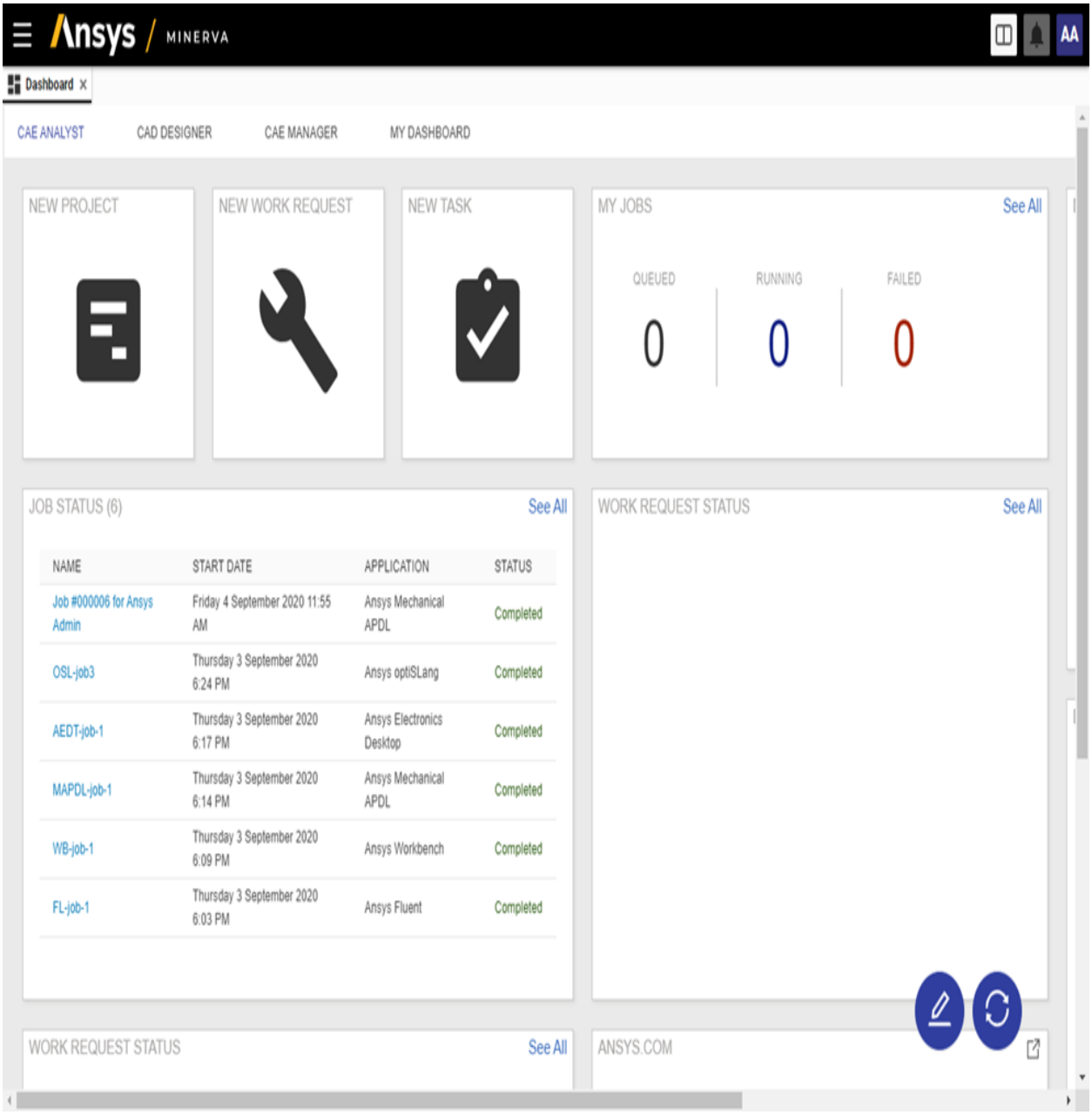
Unclaim Project: releases claimed file.

File properties: opens a dialog letting you view remote file properties. Notice the Claim Status and Claimed By properties.

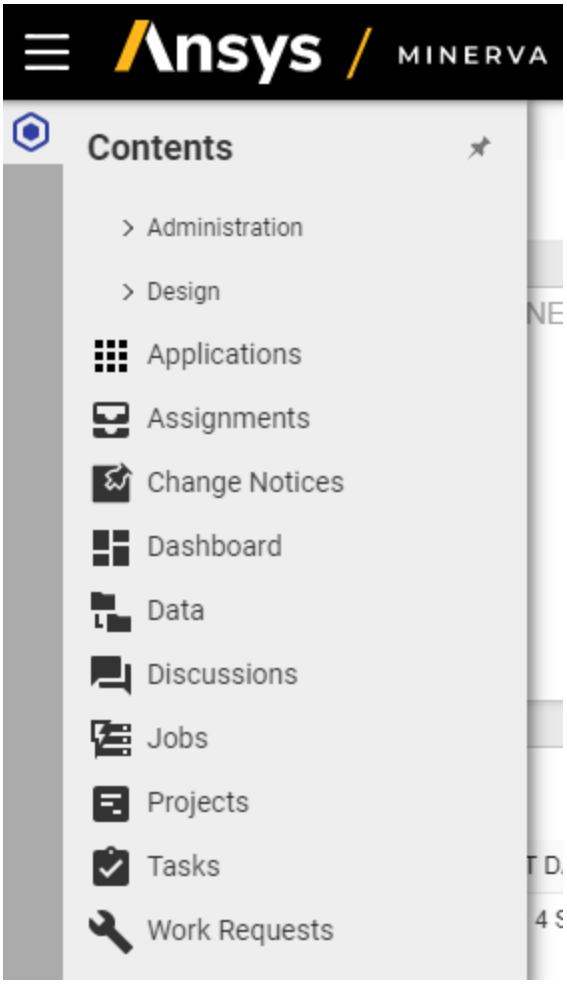


Launch Web Client:

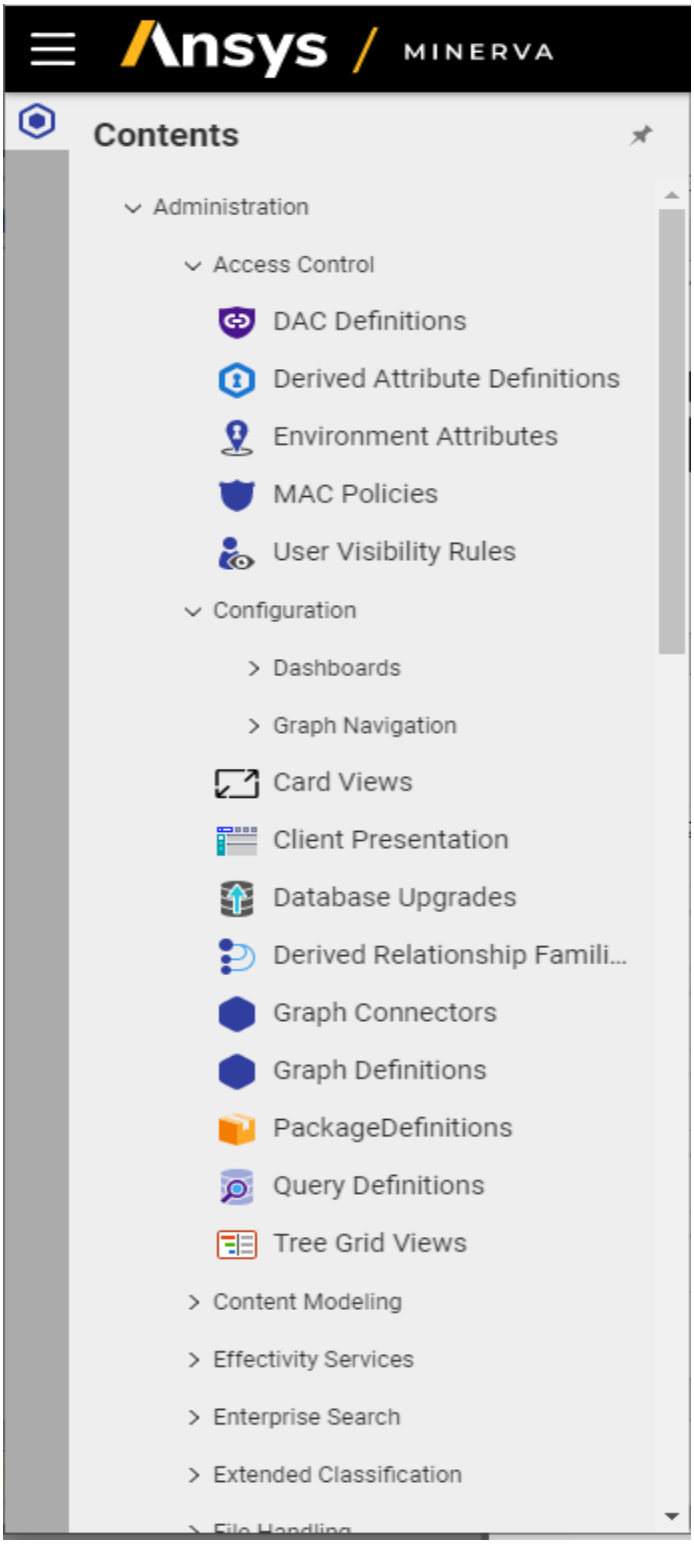
Ansys Minerva, powered by Aras, is a knowledge management application that secures critical simulation data, and provides simulation process and decision support to simulation teams across geographies and functional silos. Launching the Minverva Web Client gives access to these features. This opens a dialog calling for you to provide login information. It attempts to open a web browser window, so your default browser must be compliant.



Clicking on the triple bar icon left of the Ansys logo opens up access to the more robust Minerva Web Client functionality.



The angle icons for Administration open up further functionality.



6 - Setting Up a Maxwell Design

After you insert a design, you do not need to perform the basic steps sequentially, but they all must be completed before a solution can be generated.

To set up a Maxwell design, follow this general procedure:

1. Insert a Maxwell design into a project. After you insert the design, you can then set the model's units of measurement and the background material.
2. Draw the model geometry.
3. Specify the solver type.
4. Assign material characteristics to objects.
5. Assign boundaries and excitations.
6. (*For Transient Solutions Only*) Set up motion.
7. Add parameters for which you want to solve.
8. Specify mesh settings.
9. Specify how Maxwell will compute the solution.
10. (Optional) Set up any optimetrics you want to run.
11. Run the simulation.
12. View solution results, post-process results, view reports, and create field overlays.
13. (Optional) Export the circuit to generate a circuit equivalent of the model.

Inserting a Maxwell Design

The first step in setting up a Maxwell Project is to add a design to the active project.

To insert a design:

1. Click **Project and select one of the following:**
   - **Insert Maxwell 3D Design**
   - **Insert Maxwell 2D Design**
   - **Insert RMxprt Design**
2. The new design is listed in the project tree. It is named Maxwell3DDesign*n*, Maxwell2DDesign*n*, or RMxprtDesign*n* by default, where *n* is a number signifying the order in which the design was added to the project.

   The **Modeler** window appears to the right of the Project Manager. You can now create the model geometry.

Note	Click the plus sign to the left of the design icon in the project tree to expand the project tree and view specific data about the model.

Related Topics

Setting the Project Tree to Expand Automatically

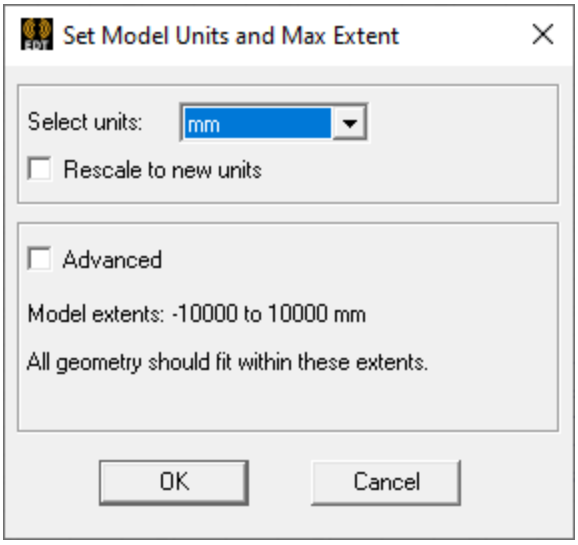
Setting Units of Measurement for the Model

You can specify the units of measurement for drawing geometric models. After the units of measurement have been specified, they are assigned to the objects in the **3D Modeler** window. You can then choose to display the model's dimensions in the new units, or rescale the model's dimensions to the new units.

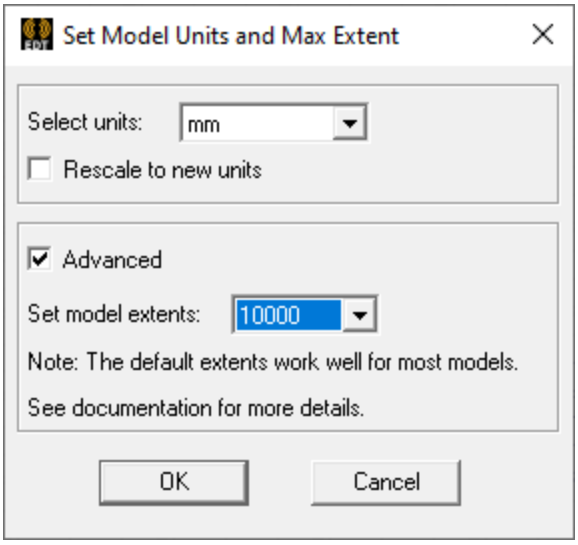
To set the model's units of measurement:

1. Click **Modeler>Units**.

   The **Set Model Units** dialog box appears.



2. Select the new units for the model from the **Select units** pull-down list.
3. Specify how the change in units affects the model:
   - Select the **Rescale to new units** option to rescale the dimensions to the new units. For example, selecting centimeters (cm) as the new unit of measurement results in a dimension of 10 millimeters (mm) becoming 10 cm.
   - Clear the **Rescale to new units** option (the default) to convert the dimensions to the new units without changing their scale. For example, selecting cm as the new unit of measurement results in a dimension of 10 mm becoming 1 cm.
4. While most models should fit the default, you can check **Advanced** to enable **Set model extents** to specify model extents in the 3D modeler.



Parasolid kernel has a strict size limit of +/- 500 units. As many AEDT models go beyond this limit in the desired units, we have implemented internal scaling to allow creation of models with a larger extent. With our default model extent of 10,000, all geometry will be internally scaled down by a factor of 100 to fit within parasolid size limit. We recommend using the default extent of 10,000 as much as possible. Modeling operations will output an error if model goes beyond 50,000 units.

A second extent option of 100 is available in the advanced options.

If model extent is set to 100, we do not scale geometry internally. Modeling operations will output an error if model goes beyond +/- 500 units. Extent of 100 should be only used in rare situations where it is determined that scaling is causing a failure, and there is no workaround.

5. Click **OK** to apply the new units to the model.

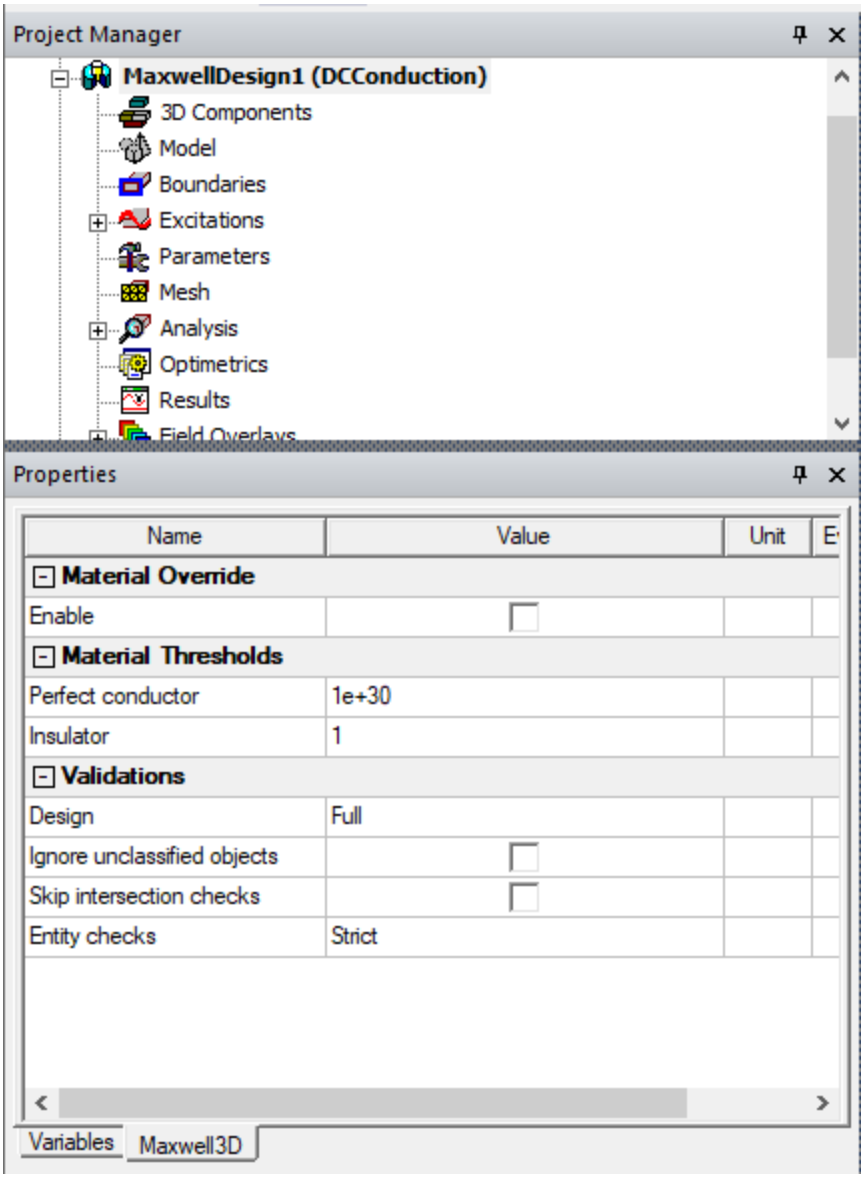
Design Settings

The Design Settings dialog allows you to specify how the simulator will deal with some aspects of the design. The tabs present in this dialog box, and the options available on each tab, vary with the solution type.

- Set the Material Thresholds for treating materials as conductors/insulators.
- Set the Model Depth Settings
- Set the Material Override (Maxwell3D only) to allow metals to automatically override dielectrics.
- Set the Matrix/Permeability Computation method for 3D Magnetostatic Solutions.
- Set the Inductance Computation method for 2D Magnetostatic Solutions
- Set the Matrix Computation to compute the inductance matrix for Transient solutions.
- Adjust model and design Validations settings.

- Set the Symmetry Multiplier for Transient and Eddy current solutions.
- Set Preserve Transient Solution options (Transient solutions Only).
- Set transient coupling with Twin Builder on the Advanced Product Coupling tab (Transient solutions Only).
- Set the Model Settings (Maxwell2D XY Transient, and 2D Eddy current solutions Only).
- Set Skew Model options (Maxwell2D XY Transient solutions with cylindrical rotational motion Only).
- Set the default Background material (Maxwell2D solutions only).
- Set Export Options (RMxprt Only).

You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.



Set Material Threshold Tab

Maxwell3D and Maxwell2D Material Thresholds:

The material thresholds tell the **Maxwell 2D** and **Maxwell 3D** solvers how to deal with conductors and insulators. Materials with conductivities above the **Perfect Conductor** threshold are treated as having infinite conductivity and surface current only. Materials with conductivity values below the **Insulator/Conductor** threshold are treated as insulators with no conductivity and no current carrying capability. Materials that fall between the two thresholds are treated as normal conductors that can carry current throughout the volume of the material.

1. Click **Maxwell>Design Settings.** The **Design Settings** dialog box appears.
   The material thresholds are set in the **Set Material Thresholds** tab. You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.
2. Type a value in the **Perfect Conductor** text box.
3. Type a value in the **Insulator/Conductor** text box.
4. Click **OK**.

RMxprt Material Thresholds:

1. Click **RMxprt>Design Settings.**
   The **Design Settings** dialog box appears with the **Set Material Threshold** tab selected. You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.
2. Type a value in the **Conductivity Threshold** text box (Default=10,000).
3. Type a value in the **Permeability** text box (Default=100).

Note	RMxprt will treat materials with conductivity greater than 10,000 as conductors, and materials with Permeability greater than 100 as steels.

4. If you want these values to be the default, change the values by clicking the **Tools>Options>RMxprt Options** menu and setting the material thresholds in the **RMxprt Options** dialog.
5. Click **OK**.

Related Topics

Setting Maxwell 3D Options

Setting Maxwell 2D Options

Setting RMxprt Options

Model Depth Settings

Maxwell 2D allows you to set the model depth for 2D magnetostatic, transient, and eddy current solutions. The model depth setting is retained when switching between magnetostatic and transient and eddy solutions.

To set the model depth:

1. Click **Maxwell>Design Settings**.
   - Alternate method: In the project tree, right-click **<***Design_name***>**, and select **Design Settings**.

     The **Design Settings** dialog box appears.

     > **Note:** You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.

2. On the **Model Depth** tab, enter the model depth value and select the desired unit of measure. The model depth value must be greater than zero. The default setting is 1 meter.

   > **Note:** Changing the model depth invalidates the solution.

3. Optionally, if you check **Save as default**, newly inserted designs will be set to use the specified model depth setting.
4. When finished changing settings, click **OK**.

Related Topics

Design Settings

Set Material Override Tab

The **Maxwell3D>Design Settings** command brings up a dialog with text note and a check box to **Enable material override**. You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.

Normally, the modeler considers any intersection between 3D objects to be an error. But, if you check this option, the modeler allows a metal object to intersect a dielectric, and just gives a warning. Intersections between two metals or two dielectrics will still be errors.

In the mesher, the dielectrics are "subtracted" from the metals in the intersecting region. That is, the part of the dielectric that is inside the metal is removed, and if the dielectric is completely inside, the whole object disappears.

The purpose of this feature is to allow you to avoid doing explicit subtraction in the modeler. One example application is a via that passes through many dielectric layers--with the option turned on, the via does not have to be subtracted from the layers.

Note	Users must be careful: this setting changes the "ground rules" of the modeler, and may have unexpected results.

Related Topics

Analyze Objects

Analyze Interobject Misalignment

Analyze Surface Mesh

Healing an Imported Object

Materials

Matrix/Permeability Computation Tab Settings for 3D Magnetostatic Solutions

Maxwell 3D provides two **Matrix/Permeability Computation** methods for calculating the inductance of current-carrying coils for magnetostatic solutions: **Apparent** and **Incremental**.

- **Apparent** inductance is defined as the ratio of flux to current at a certain operating condition. Alternatively, this can be expressed as the slope of a line from the origin to the operating point on the BH curve. Apparent inductance is commonly used in nonlinear magnetic applications and in circuit analysis, and is the default method used by Maxwell.
- **Incremental** inductance is defined as the ratio of delta flux to delta current at a certain operating condition. This is equivalent to the slope of the tangent to the BH curve at the operating point. Incremental inductance is commonly used in physical design because it is easy to measure.

To set the matrix/permeability computation method:

1. Click **Maxwell>Design Settings**.
   - Alternate method: In the project tree, right-click **<***Design_name***>**, and select **Design Settings**.

   The **Design Settings** dialog box appears. You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.
2. On the **Matrix/Permeability Computation** tab, select the desired method: Apparent or Incremental.

Note	If you change the computation method after solving, all matrix results of both AdaptivePass and LastAdaptive solutions will be invalidated. Then, on the next simulation, the matrix of the last solved pass will be re-computed with the new method. If there are matrix expression caches for the AdaptivePass solution, then the cached values of previous passes will not be available because only the last pass is re-simulated.

3. Optionally, if you check Save as default, newly inserted designs will be set to use the specified computation method.
4. When finished changing settings, click **OK**.

Related Topics

Design Settings

Using the Permeability Link Option for Eddy Current Solutions

Inductance Computation Tab Settings for 2D Magnetostatic Solutions

Maxwell 2D provides two **Inductance Computation** methods for calculating the inductance of current-carrying coils for magnetostatic solutions: **Apparent** and **Incremental**.

- **Apparent** inductance is defined as the ratio of flux to current at a certain operating condition. Alternatively, this can be expressed as the slope of a line from the origin to the operating point on the BH curve. Apparent inductance is commonly used in nonlinear magnetic applications and in circuit analysis, and is the default method used by Maxwell.
- **Incremental** inductance is defined as the ratio of delta flux to delta current at a certain operating condition. This is equivalent to the slope of the tangent to the BH curve at the operating point. Incremental inductance is commonly used in physical design because it is easy to measure.

To set the matrix computation method:

1. Click **Maxwell>Design Settings**.
   - Alternate method: In the project tree, right-click **<***Design_name***>**, and select **Design Settings**.

   The **Design Settings** dialog box appears. You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.
2. On the **Inductance Computation** tab, select the desired method: Apparent or Incremental.

Note	If you change the computation method after solving, all matrix results of both AdaptivePass and LastAdaptive solutions will be invalidated. Then, on the next simulation, the matrix of the last solved pass will be re-computed with the new method. If there are matrix expression caches for the AdaptivePass solution, then the cached values of previous passes will not be available because only the last pass is re-simulated.

3. Optionally, if you check Save as default, newly inserted designs will be set to use the specified computation method.
4. When finished changing settings, click **OK**.

Related Topics

Design Settings

Matrix Computation Tab Settings for Transient Solutions

Maxwell 2D/3D provides an option on the **Matrix Computation** tab for computing the inductance matrix for transient solutions. When the **Compute inductance matrix** option is selected, the computed matrix quantities will be available at all time steps. The matrix quantities are listed in the reporter and thus are available for post-processing, output variables creation, and Optimetrics studies.

Note	During the inductance calculation in the transient solver: • Eddy effects for all objects are turned-off even if “Eddy Effect” is checked in the

	"Set Eddy Effects" dialog box. However, the field solution and all other results will consider the eddy effects as set in the interface. • For a field system with multiple stranded windings, after all nonlinear materials are frozen, one winding is supplied with 1A DC current while the other winding is open (all other sources are turned-off).

The size of the computed inductance matrix is equal to the number of winding setups, i.e., the matrix will be *N* x*N* when there are *N* winding setups; and the total number of inductance-related quantities will be N^2. Components of the full matrix are available for plots and Optimetrics calculation even though the matrix will always be symmetric. You can choose to access only the upper right half of the matrix by selecting the corresponding quantities as listed in the reporter dialog box. For example, the upper half of a 2x2 matrix will be the L11, L12 and L22 quantities.

Maxwell 2D/3D provides two **Matrix Computation** methods for calculating the inductance of windings for transient solutions: **Apparent** and **Incremental**.

- **Apparent** inductance is defined as the ratio of flux to current at a certain operating condition. Alternatively, this can be expressed as the slope of a line from the origin to the operating point on the BH curve. Apparent inductance is commonly used in nonlinear magnetic applications and in circuit analysis, and is the default method used by Maxwell.
- **Incremental** inductance is defined as the ratio of delta flux to delta current at a certain operating condition. This is equivalent to the slope of the tangent to the BH curve at the operating point. Incremental inductance is commonly used in physical design because it is easy to measure.

To enable the matrix computation for transient solutions:

1. Click **Maxwell>Design Settings**.
   - Alternate method: In the project tree, right-click **<***Design_name***>**, and select **Design Settings**.

   The **Design Settings** dialog box appears.You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.
2. On the **Matrix Computation** tab, select **Compute inductance matrix**, and select the desired method: **Apparent** or **Incremental**.

Note	
	• The default is NOT to compute the inductance matrix. Changing the setting for an existing design does not affect the solution for the previous simulation. For example, if a transient solve setup has 1 second set as the end time and is solved with the **Compute inductance matrix** option off (unchecked) and a plot of matrix quantities is created vs. time, the plot will not show any values because no matrix quantities were computed. Subsequently, if the compute option is enabled (checked) and the end time extended to 2 seconds, after simulation, matrix quantities will be available from > 1 second to 2 seconds, at all solved time steps. • If you change the computation method after solving, all matrix results of both AdaptivePass and LastAdaptive solutions will be invalidated. Then, on the next simulation, the matrix of the last solved pass will be re-computed with the new method. If there are matrix expression caches for the AdaptivePass solution, then the cached values of previous passes will not be available because only the last

	pass is re-simulated.

3. Optionally, if you check **Save as default**, newly inserted designs will use the specified setting.
4. When finished changing settings, click **OK**.

Related Topics

Design Settings

Inductance Computation for 2D and 3D Transient Solutions

Validations Tab

Model Settings

You can adjust the degree to which the software checks a model for faults that could jeopardize mesh accuracy. There are four levels of model validation that a user can specify for a given design: **Warning Only**, **Basic**, **Strict**, and **None**. Note that this setting affects only the "3D Model" stage of a design validation. The default Entity Check Level is **Strict**.

- The **Warning Only** entity check setting allows all models to pass 3D Model validation regardless of any faults that are found. These faults are posted in the message window as warnings.
- The **Basic** entity check setting allows most models to pass 3D Model validation. Some faults are flagged as model errors, thereby prohibiting a design from proceeding to the meshing stage of an analysis. You must correct such errors before attempting to analyze the design under the Basic setting, or you must change the entity check level to Warning Only.
- The **Strict** entity check setting enforces a tighter tolerance for model faults than the Warning Only and Basic settings. All model faults that are found during 3D Model validation are posted to the message window. These errors must be corrected before attempting to analyze the design under the Strict setting, or you must change the entity check level to Basic or Warning Only.
- The **None** setting disables model validation.

In addition, check boxes allow you to adjust model validation to:

- **Ignore Unclassified Objects** and to
- **Skip Intersection Checks**.

These settings are unchecked by default.

Maxwell2D and Maxwell3D Design Settings

For both 2D and 3D designs, you can choose either:

- **Perform full validations** (the default setting) – All design validations are performed.
- **Perform minimal validations** – All design validations except boundary overlap validation are performed.

For 3D designs only, you can select **Skip mesh checks** to prevent the solver from failing validation when the mesh has a few elements of poor quality.

Adjusting Settings

To adjust settings on the Validations tab:

1. Select **Maxwell>Design Settings** to open the **Design Settings** dialog box; then select the Validations tab.
2. Choose the desired level of Model validation from the **Entity Check Level** drop down box.
3. Optionally, adjust Model validation to **Ignore Unclassified Objects**, or to **Skip Intersection Checks**.
4. Optionally, select **Perform minimal validations** if you want to omit boundary overlap validation checking.
5. Optionally, **Skip mesh checks** to prevent the solver from failing validation when the mesh has a few elements of poor quality.
6. Optionally, click the **Save as Default** button to make the current selections the default settings.
7. Click **OK** to accept the selections and close the dialog.

> **Note:** You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.

Related Topics

Model Analysis

Analyze Objects

Interobject Misalignment

Analyze Surface Mesh

Heal

Technical Notes: *Healing and Meshing*

Technical Notes: *Detecting and Addressing Model Problems to Improve Meshing*

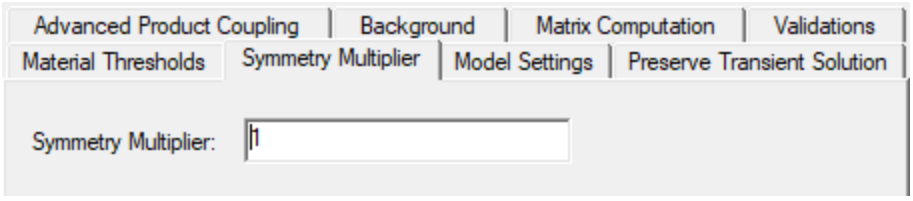
Setting a Symmetry Multiplier

(Transient and Eddy Current Solutions Only)

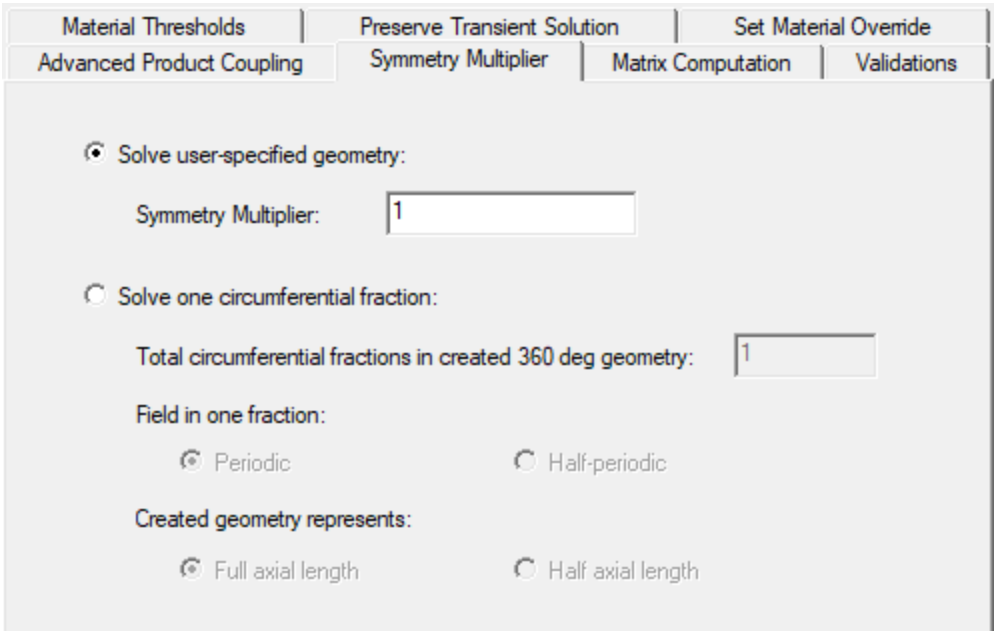
If you are using symmetry and/or independent/dependent boundary conditions in your solution, you may need to specify a multiplier (so that the other part of the model, which is not included in the solution domain, can be properly taken into account). This symmetry multiplier will be automatically applied to all input quantities including: input voltage, inductance, resistance, load torque, mass, damping, external circuit; and all output quantities including: induced voltages, flux linkages in every winding, stranded loss, solid loss, core loss, and torque.

To specify a multiplier:

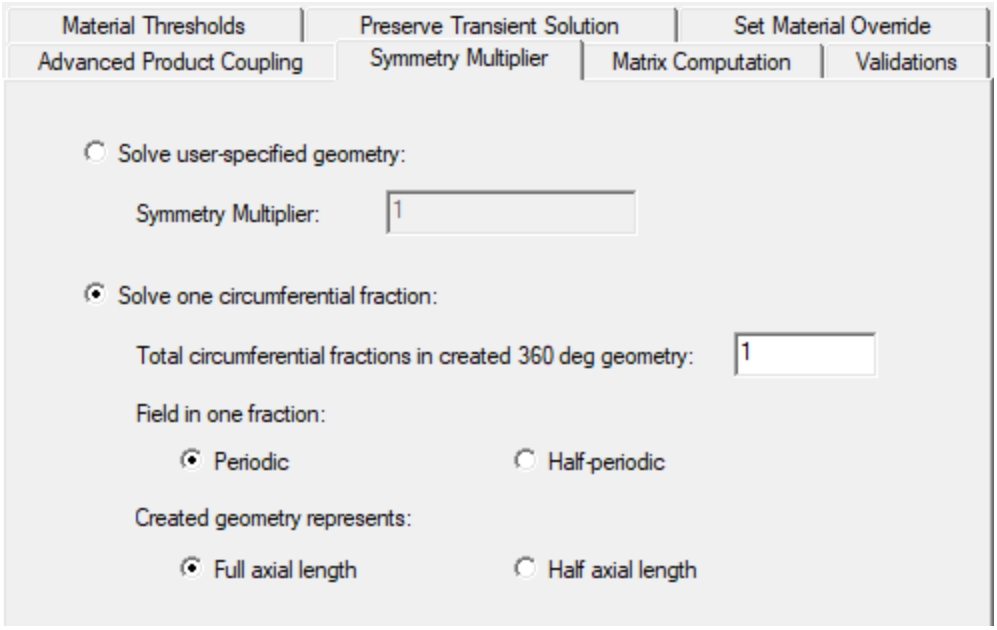
1. Select **Maxwell>Design Settings** from the main menu to open the Design Settings dialog box, then select the **Symmetry Multiplier** tab. You can also right-click **Model** in the Project Manager tree, and select **Set Symmetry Multiplier**; or select **Maxwell>Model>Set Symmetry Multiplier**, to open the Design Settings dialog to the **Symmetry Multiplier** tab.You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.
2. Depending on the design type, do one of the following:
   - For 2D and 3D Eddy Current, and 2D Transient designs, enter an integer value (or a variable that evaluates to an integer value) in the **Symmetry Multiplier** text box, and click **OK**.



   - For 3D Transient designs, you can select either:
     - **Solve user-specified geometry** and enter an integer value (or a variable that evaluates to an integer value) in the **Symmetry Multiplier** text box, then click **OK**.



- **Solve one circumferential fraction**, then enter an integer value for the **Total circumferential fractions in created 360 deg geometry**.

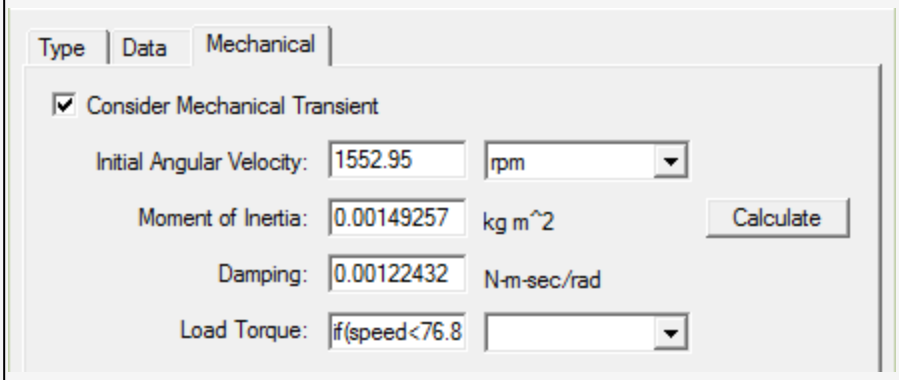


Select either **Full axial length** or **Half axial length** for **Created geometry represents**.

Select either **Periodic** or **Half-periodic** for **Field in one fraction**.

Click **OK**.

Note	When a symmetry multiplier has been set, the parameter values (moment of inertia, damping, load torque) specified on the **Motion Setup** dialog **Mechanical** tab are related to the full model.



Related Topics

Modifying the Model View

Assigning Boundaries and Excitations for 3D Designs

Assigning a Symmetry Boundary for a Transient Solver

Assigning a Symmetry Boundary for the Eddy Current Solver

Assigning a Symmetry Boundary for a 2D Transient Solver

Assigning a Symmetry Boundary for the 2D Eddy Current Solver

Export Options Tab

You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver. To set the Export options for RMxprt:

1. Choose which options to use when exporting a design from RMxprt to Maxwell3D:
   - **Periodic**
   - **Difference**
   - **Band Arc**
   - **Teeth-Teeth**
   - **Segmented Arc**

Note	These options may also be set on the **Export Options** tab of the RMxprt Options dialog box. Using the **Tools>Options>RMxprt Options** command changes the default for the current design and all future designs.

You may also set the default **Design Sheet** for use with RMxprt by entering the path and filename or by browsing to the Excel file using the **ellipsis** button.

Related Topics

Generating a Custom Design Sheet for RMxprt

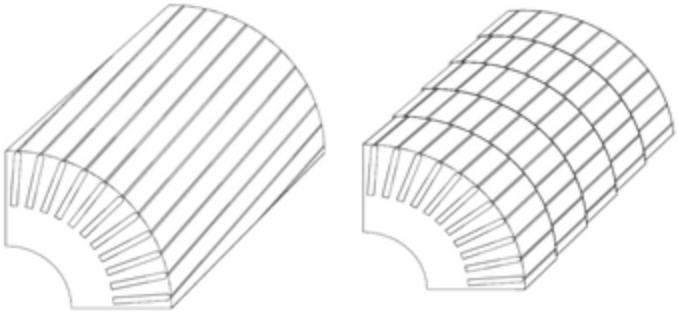
Model Settings Tab

(Maxwell2D XY Transient and 2D Eddy Current Designs Only)

For Maxwell2D XY transient and 2D Eddy current designs, torque, flux linkage, back EMF, etc. are scaled by the length provided in the **Model Depth** field on the **Model Settings** tab. The scaled results are used to calculate output quantities.

(Maxwell2D XY Transient solutions with cylindrical rotational motion Only)

For Maxwell 2D transient designs with cylindrical rotational motion setup, the transient solver can take into account the effects of skewed rotor slots by using a multi-slice model.

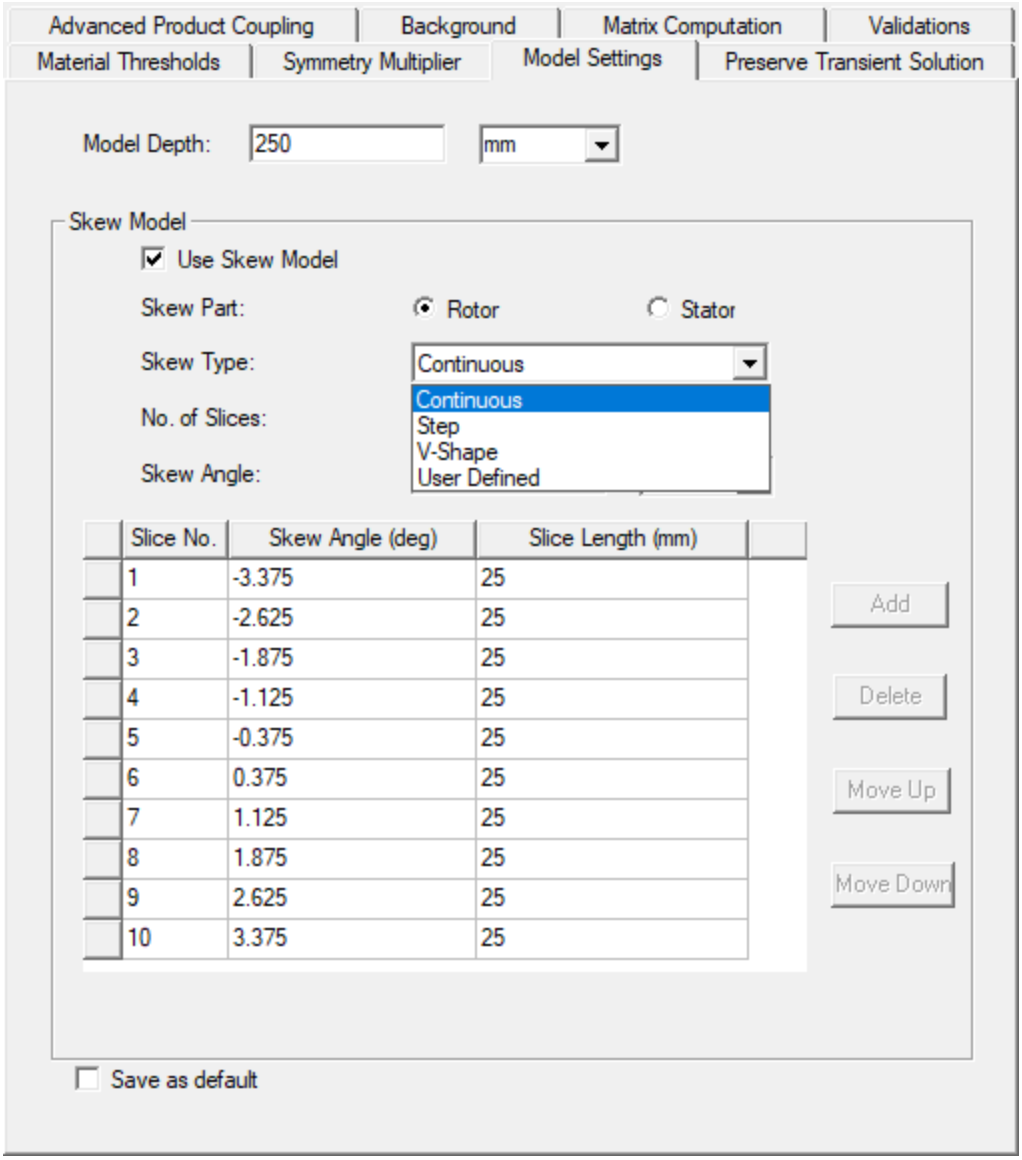


Skewed rotor Multi-slice model

Note: Maxwell 2D transient designs that use the skew model can use MPI HPC for distributed computing of multiple slices.

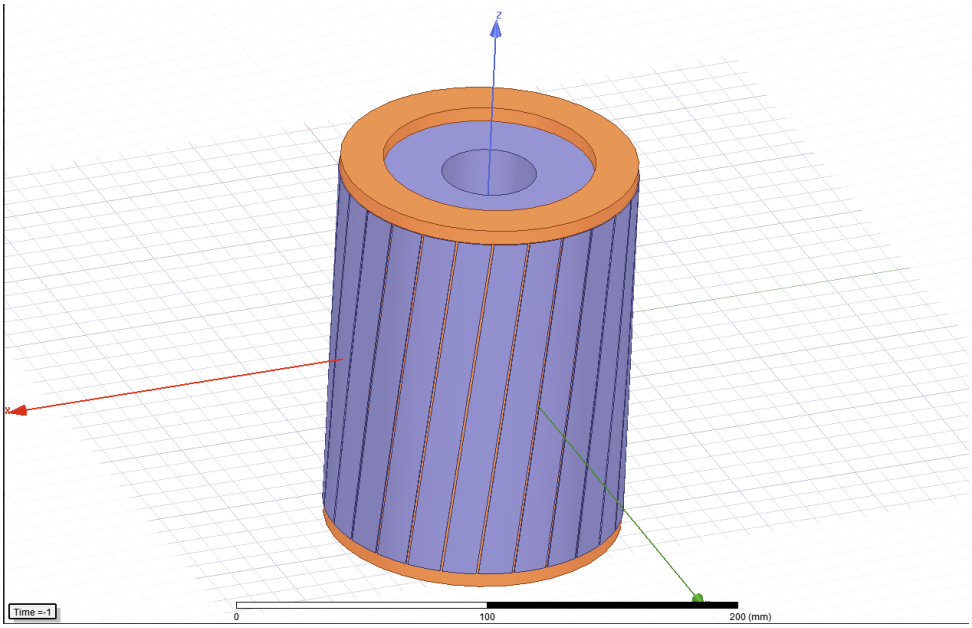
To enable use of the multi-slice skew model:

1. Right-click on **Model** in the Project Tree and select **Set Skew Model** on the context menu to open the **Design Settings** dialog to the **Model Settings** tab. You can also click **Maxwell 2D> Design Settings** to open the **Design Settings** dialog box, then select the **Model Settings** tab. You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.
2. Check the **Use Skew Model** check box to enable the parameter fields below it.

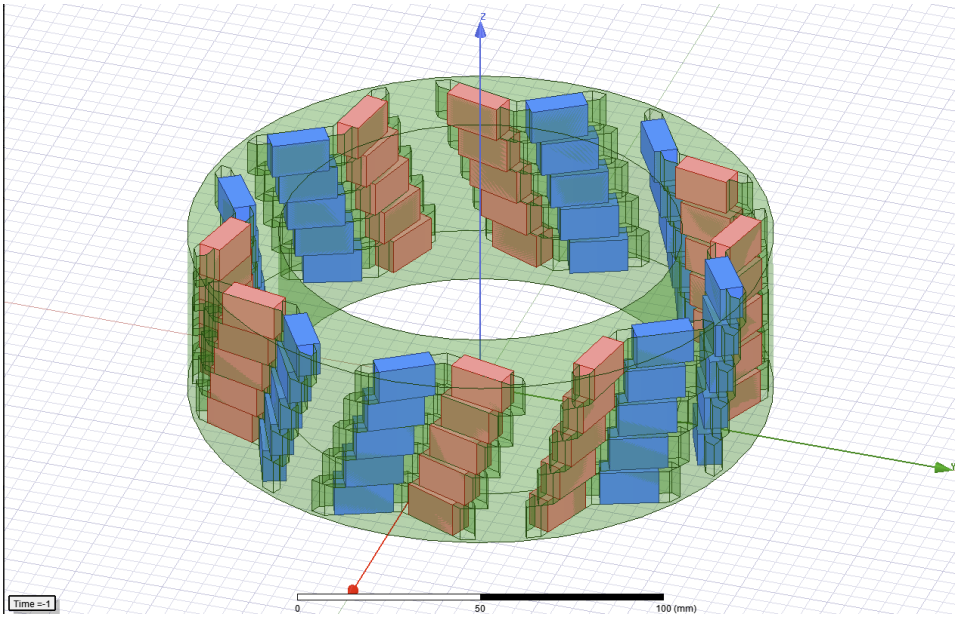


3. Select either **Rotor** or **Stator** as the **Skew Part** being modeled. Default is **Rotor**.
4. Select a **Skew Type**.

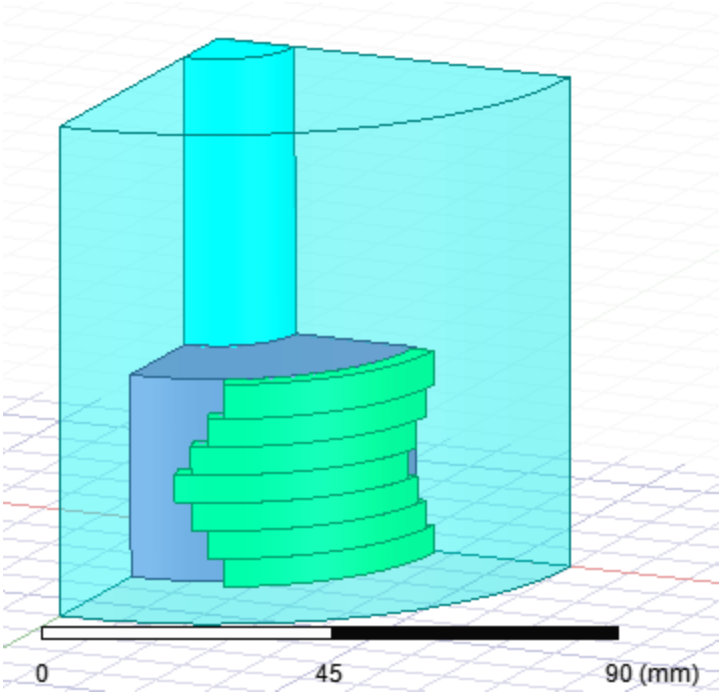
- **Continuous** (the default) - usually used in induction machines



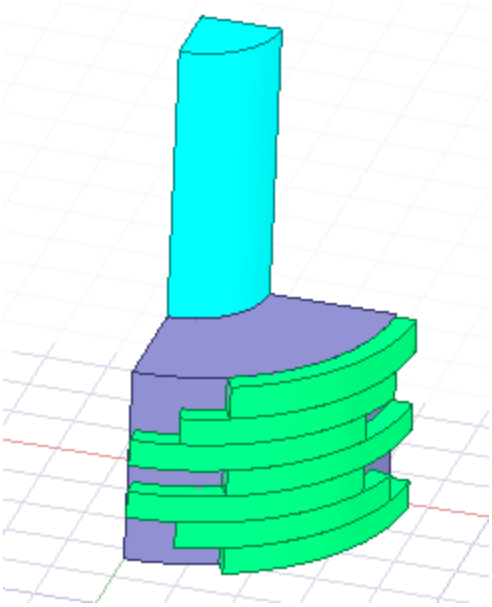
- **Step** - usually used in PM machines



- **V-Shape**

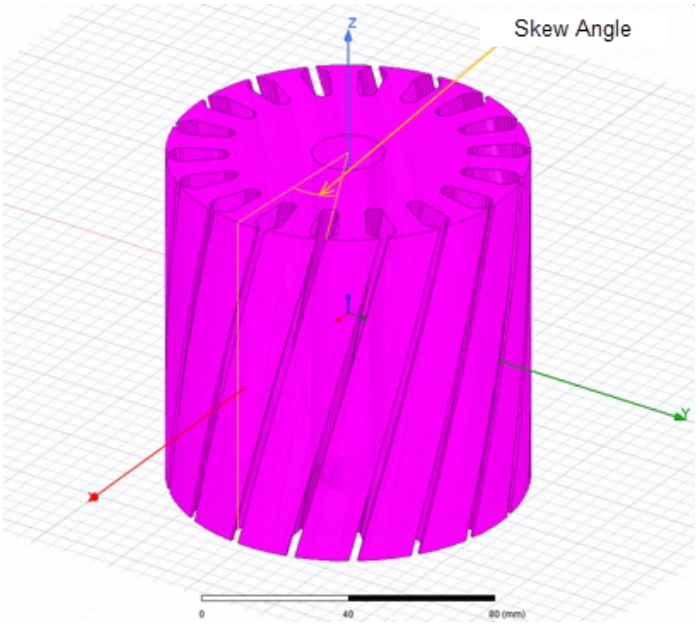


- **User Defined**

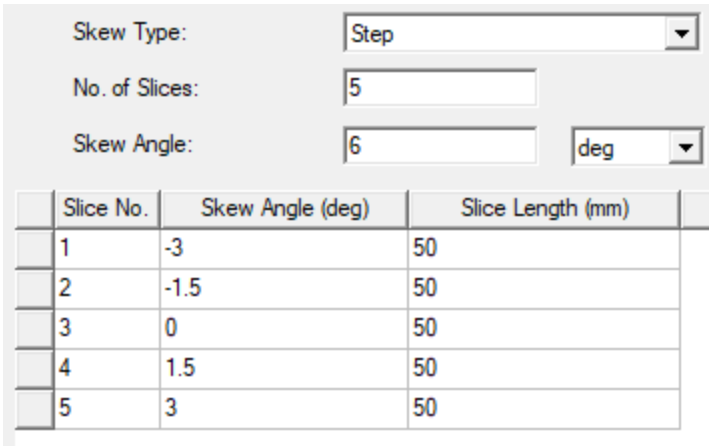


5. For **Continuous**, **Step**, and **V-Shape** skew types:
    a. Enter the **No. of Slices** that you want the model divided into in the Z-direction. The input must be a positive integer value greater than zero. The default is 5. You can also enter a variable name. (**No. of Slices** is not used for the **User Defined** skew type and is grayed out .)

b. Enter the **Skew Angle** and unit of measure of the part. The **Skew Angle** is the angle between first and last slice (not between two adjacent slices). Entries are non-negative real numbers in the range of 0 to 180 degrees inclusive (or equivalent values in units of measure such as radians). The default value is 5 degrees. You can also enter a variable name.

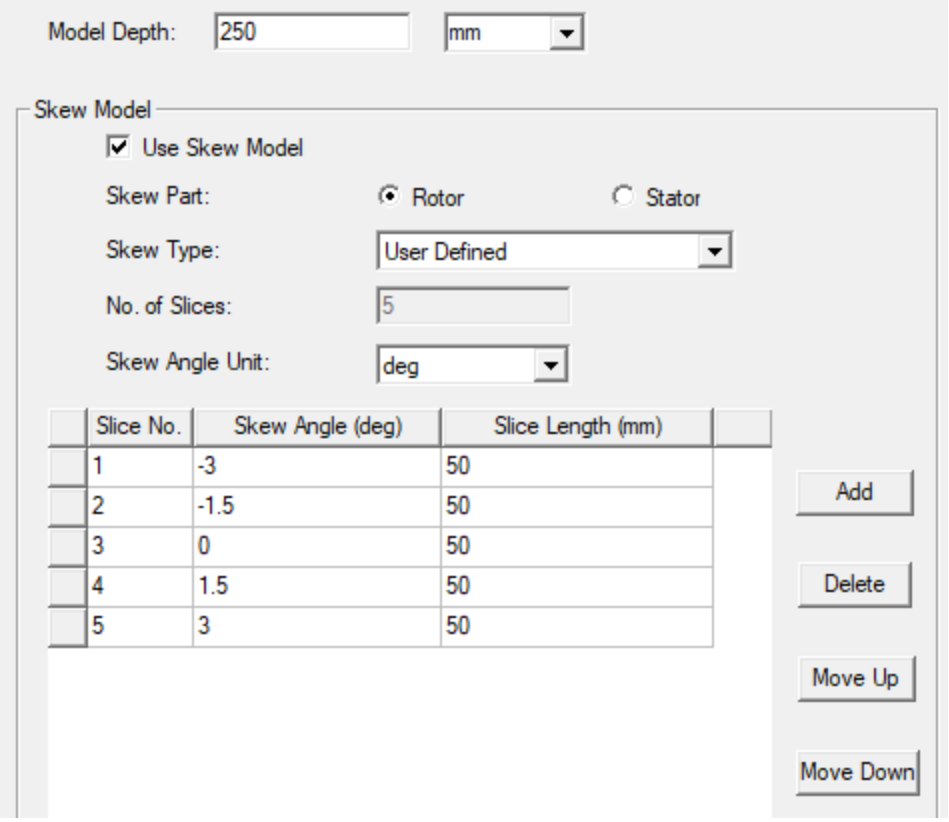


The Skew Angle and Slice Length values for each slice are calculated by the modeler and are listed in the (non-editable) table.



Slice No.	Skew Angle (deg)	Slice Length (mm)
1	-3	50
2	-1.5	50
3	0	50
4	1.5	50
5	3	50

6. For the **User Defined** skew type:
   a. Select a **Skew Angle Unit**, which will be applied to the Skew Angle values you enter in the table.



b. Enter the desired **Skew Angle** and **Slice Length** values for the selected skew part. If the total of the **Slice Length** values differs from the **Model Depth** value, a message informing you of the difference will display in the Message Manager. **Add**, **Delete**, **Move Up**, and **Move Down** buttons allow you to control the table row entries.

7. When finished, click **OK** to close the dialog box.

Related Topics

Harmonic Force Calculation in Maxwell 2D with Skew Model

User Defined Skew Type Computations

The phase angle of flux-linkage in a user defined skew part will change according to the slice skew angle.

For user defined skew type, solver will rotate all the slices at an additional angle β in order to keep flux-linkage with the same initial phase angle as that of the unskewed case.

The additional angle β can be calculated by

$$\beta = -\frac{1}{L}\sum_{i=1}^{i=N} \alpha_i * L_i$$

where:

- β is the additional rotate angle for all the slices
- N is the slice number
- α_i is the ith slice skew angle which user defines
- L_i is the ith slice length which user defines
- $L = \sum_{i=1}^{i=N} L_i$ is the model depth.

Thus, the slice skew angle for each slice will be:

$\alpha_i + \beta$

Related Topics

Background Material Tab

(Maxwell2D Designs Only)

The area of a model not enclosed within specific objects is termed the background. The material assigned to this area is vacuum by default.

To change the default setting:

1. Click **Maxwell2D>Design Settings**.

   The Design Settings dialog is displayed. You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.
2. Select the **Background** tab.
3. Click the **Background** button listing the current material setting.

   The **Select Definition** dialog is displayed with the Material Database.
4. Locate the material desired and click on the **Name** in the table.
5. Click **OK** to return to the **Design Settings** dialog.
6. The **Background** button will now list the newly selected material indicating that it will be used as the background material.
7. Click **OK** to finalize the change and dismiss the dialog.

Preserve Transient Solution Tab

For **Transient** solutions, you have the option of enabling the **Preserve solution after dataset edits** function on this tab.

Datasets may be used to describe material and source behavior in Maxwell. Enabling the **Preserve solution after dataset edits** function allows source, material, or other parameters that depend upon datasets to be modified without deleting the transient solution data. The transient solver may then be restarted from the final time step of the previous simulation.

You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.

Advanced Product Coupling Tab

For Transient solutions, this tab provides the option to **Enable transient-transient link with Twin Builder** components. You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.

Maxwell designs may be dynamically coupled to Twin Builder components through the Twin Builder user interface. The Twin Builder interface allows 2D, 3D, and RMxprt designs to be linked as dynamic components. Source windings set to **External** are available as conservative pins in Twin Builder.

Dynamically linked components provide the ability to have Twin Builder vary a variable in a Maxwell design and have Maxwell solve the design at the new design point and provide the solution data back to Twin Builder for use in the circuit simulation.

For more information, refer to the Twin Builder Help.

Related Topics

Assigning a Winding Setup for a Transient Solver

Assigning a Winding Setup for a 2D Transient Solver

7 - Specifying the Solver Type

Before you begin to draw the models in your project, choose which field solver to use for it. Each solver requires a different type of problem setup. If you later change the solver, all problem setups become invalid, and all solutions are deleted. Because of this, it is a good idea to decide on a solver before starting.

To specify a solver:

1. Click **Maxwell 2D>Solution Type** or **Maxwell 3D>Solution Type**.
   The **Solution Type** window appears with the solvers generally split between **Magnetic** and **Electric** solver types.
2. For Maxwell 3D designs, select one of the following solver types:
   - **Magnetostatic**
   - **Eddy Current**
   - **Transient**
   - **Electrostatic**
   - **3D AC Conduction**
   - **DC Conduction** either with or without Insulator Field
   - **Electric Transient**
3. For Maxwell 2D designs, select one of the following solver types:
   - **Magnetostatic**
   - **Eddy Current**
   - **Transient**
   - **Electrostatic**
   - **2D AC Conduction**
   - **DC Conduction**

Note	For Maxwell2D designs, you must also specify the geometry mode, either **Cartesian, XY** or **Cylindrical about Z**.
4. Click **OK**.

Note	For information on the various material property types supported by each solver, refer to *Maxwell Solvers Material Properties*.

Related Topics

Disconnected Solution Domains

Specifying Solution Settings

Post-Processed Quantities

Maxwell Solvers Material Properties

Magnetostatic Solver

The magnetostatic field simulator computes static magnetic fields in 2D or 3D. The source of the static magnetic field can be:

- DC currents in conductors.
- Static external magnetic fields represented by boundary conditions.
- Permanent magnets.

The quantities for which the magnetostatic field simulator solves are the magnetic field, **H**, and the current distribution, **J**; the magnetic flux density, **B**, is automatically calculated from the **H**-field. Derived quantities such as forces, torques, energy, and inductance may be calculated from these basic field quantities.

Related Topics

Technical Notes: 3D Magnetostatic Field Calculation

Technical Notes: 2D Magnetostatic Field Simulation

Technical Notes: Magnetostriction Modeling of Magnetic Material

Magnetostriction and Inverse Magnetostriction

Eddy Current Solver

The eddy current (AC magnetics) field simulator computes electromagnetic fields in the frequency domain in 2D or 3D. The sources of the electromagnetic field can be:

- AC currents in conductors.
- Time-harmonic external magnetic fields represented by boundary conditions.

The quantity for which the eddy current field simulator solves is the magnetic field, **H**. The magnetic flux density, **B**, is automatically calculated from the **H**-field. Optionally, the eddy current solver can incorporate displacement currents into the solution even in non-conducting domains, thereby allowing the solution of the full system of Maxwell's equations and the recovery of the electric field **E** everywhere including the non-conducting regions. Derived quantities such as forces, torques, energy, losses, and impedances may be calculated from these basic field quantities at the solution frequencies.

Related Topics

Technical Notes: 3D Frequency Domain (Eddy Current) Solver

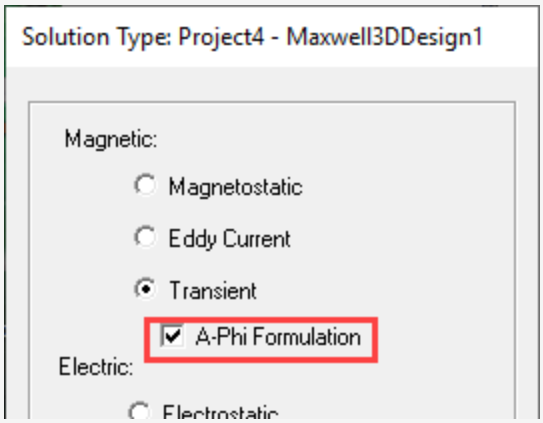
Technical Notes: 2D Eddy Current Field Simulation

Transient Solver

The transient field simulator computes the time-domain magnetic fields in 2D or 3D. The source of the magnetic fields can be:

- Moving or non-moving time varying currents and voltages.
- Moving or non-moving permanent magnets and/or coils.
- Moving or non-moving external circuit coupling.

The quantities for which the transient field simulator solves are the magnetic field, **H**, and the current distribution, **J**; the magnetic flux density, **B**, is automatically calculated from the H-field. Derived quantities such as forces, torques, energy, speed, position, winding flux linkage, and winding induced voltage may be calculated from these basic field quantities.

Note	If **A-Phi Formulation** is selected as the 3D Transient solver, please refer to **Maxwell 3D Transient Solution Based on A-Phi Formulation** for details. 

Related Topics

Technical Notes: 3D Transient Excitations (Sources)

Technical Notes: 2D Transient Simulation

Technical Notes: Magnetostriction Modeling of Magnetic Material

Technical Notes: A-Phi Formulation in Maxwell 3D (Transient)

Magnetostriction and Inverse Magnetostriction

Maxwell 3D Transient Solution Based on A-Phi Formulation

The Maxwell 3D transient solver is currently based on T-Omega formulation. T-Omega formulation is a powerful method for solving a wide range of low frequency EM problems. However, there are some areas where the applicability of the method is limited – such as multiple (mixed) source excitations on a single conduction path and capacitive effect (displacement current). For simulation of such problems, the Maxwell 3D Transient A-Phi formulation is a more suitable solution type.

Please refer to the following sections for information on this feature:

- "Limitations" on the next page
- "Boundaries and Excitations" on the next page

Limitations

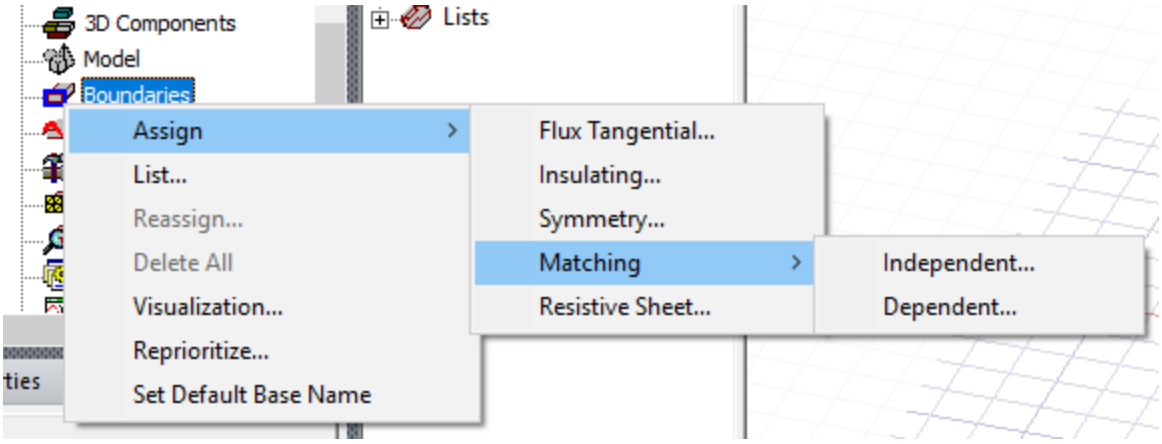
The following 3D Transient features are not supported/available for this release:

- Time Decomposition Method Option on the **Solver** tab of **Solve Setup**.
- Solver level of Auto HPC and MPI.
- Motion related features.

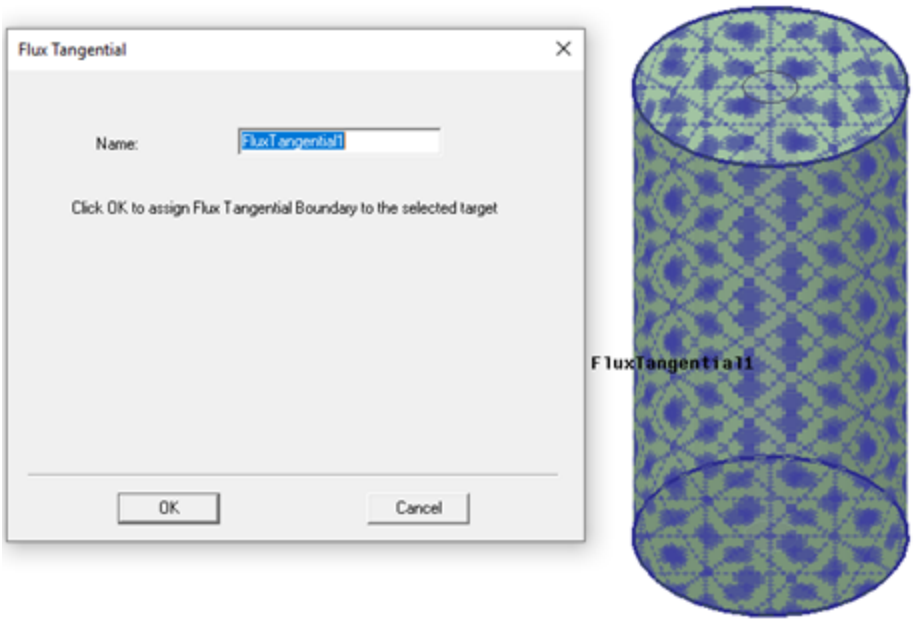
Boundaries and Excitations

Boundaries and Excitations supported are as follows:

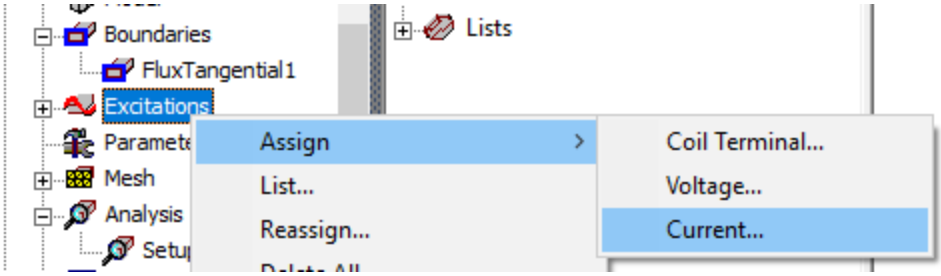
- Boundaries supported are: **Flux Tangential**, **Insulating**, **Symmetry**, **Independent**, **Dependent**, and **Resistive Sheet**.



Flux Tangential boundaries can only be applied to Face objects. Unlike T-Omega Formulation, the Neumann boundary condition on external faces is flux normal for A-Phi formulation. That is why a Flux Tangential boundary condition must be assigned when the magnetic field is tangential to the face.



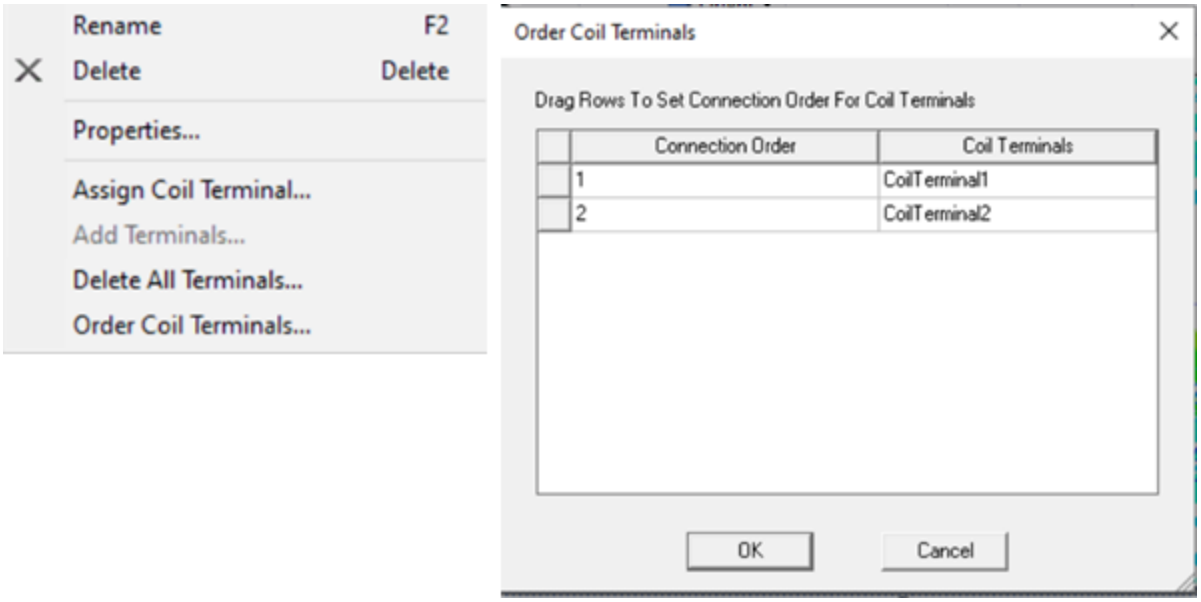
- Excitations supported are winding (via **Coil Terminal**), voltage, and current. Winding excitation is assigned as in the Transient solver (Refer to *Coil Terminals* for more information on winding assignments). **Voltage** and **Current** excitations are also included on the **Excitations>Assign** menu:



Winding Excitations for A-Phi Formulation

Winding excitation is assigned as in the Transient solver (Refer to *Assigning a Winding Setup for a Transient Solver* for more information on winding assignments). For winding excitations, the coil terminals must be ordered according to the physical potential distribution. To do this:

1. Right-click on the winding to open the context menu.
2. Click on **Order Coil Terminals** to open the **Order Coil Terminals** dialog box.



3. Order the terminals by dragging them in the **Order Coil Terminals** dialog box.

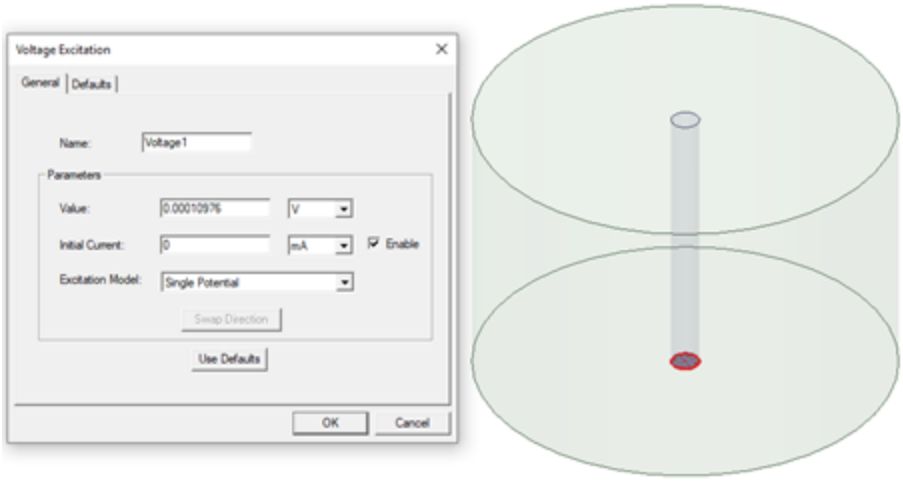
Voltage Excitations for A-Phi Formulation

For designs with known potentials, select and assign a Voltage excitation on each face.

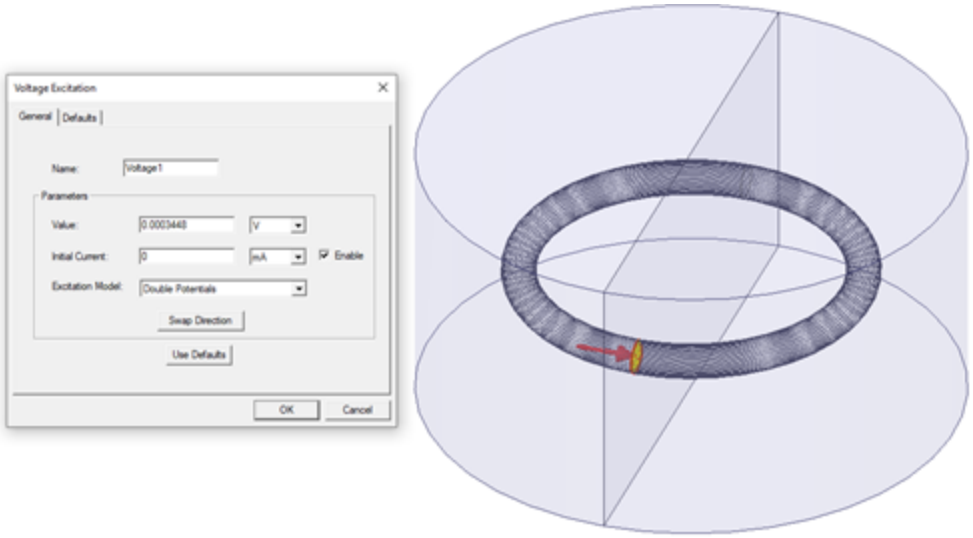
You can also define initial currents for Voltage Excitation by checking the **Initial Current** box. When Initial current is defined, the solver treats it as the Current excitation. The default value is 0A. You should select at least one voltage excitation without enabling the initial currents on a solid conduction path to set a reference potential. (If the path is touching the odd symmetry boundary, where the potential is set to 0 V, or if displacement effect is considered in the problem domain, then this rule does not apply on this path.)

Voltage excitation supports 3 different excitation models:

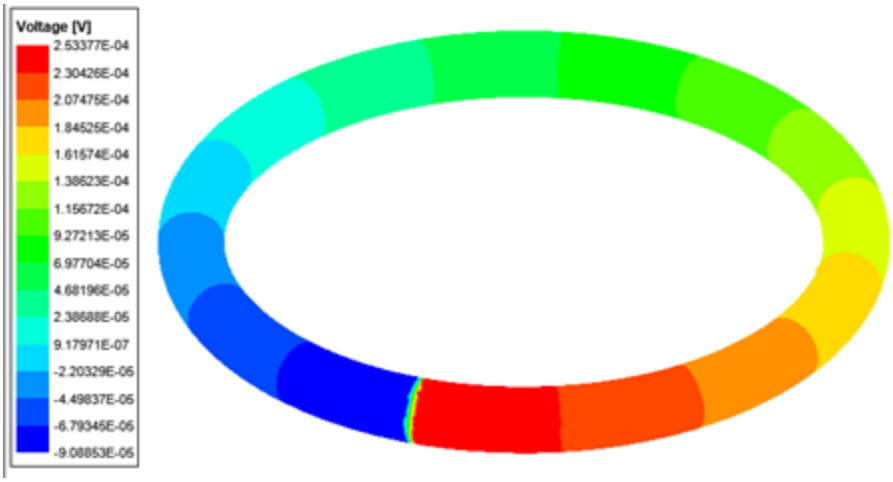
- **Single Potential**: The solver treats the terminal as an equi-potential terminal. Only one scalar electric potential DOF (Degree of Freedom) is assigned to the surface and the value of the DOF is set to the value that user inputs. *This model must be used only at the external boundaries of the problem domain.*



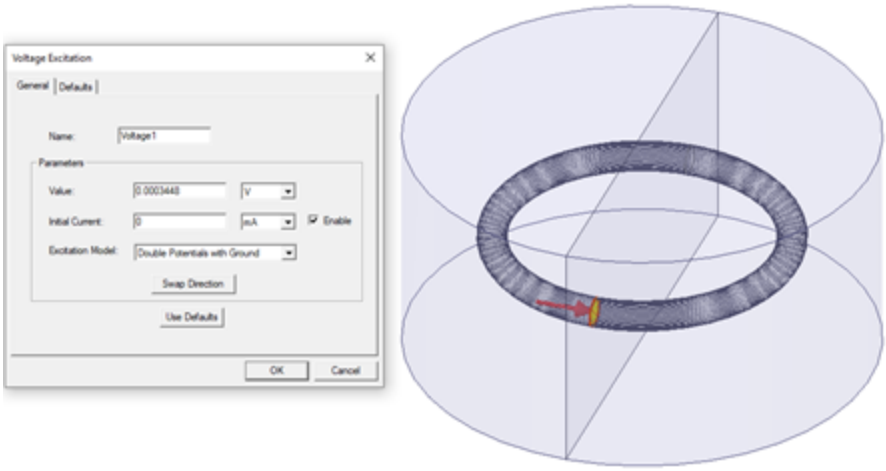
- **Double Potentials**: The solver treats the terminal as a voltage (potential difference) terminal. At the terminal face, two scalar electric potential DOFs are defined and the DOFs are unknown. The difference between those potentials equals the voltage value that the user specified. If the voltage value is a positive quantity, higher potential is in the specified direction. *This model must be used on faces inside the problem domain.*

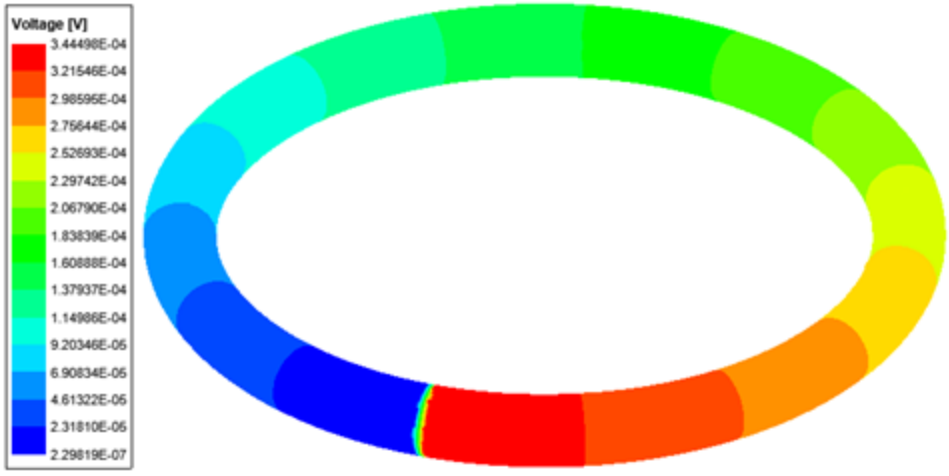


If the user does not define a reference potential on the same conduction path, the potentials at the terminal will be floating, so they are not unique. However, B, J, H, and all other quantities will be unique since they depend on the derivative of the potential.



- **Double Potentials with Ground**: The solver treats the terminal as voltage (potential difference) terminal. At the terminal face, two scalar electric potential DOFs are defined and both DOFs are known. The difference between those potentials equals to the voltage value that user specified. If the voltage value is a positive quantity, the DOF in the specified direction is set to the specified voltage value, and the DOF in the opposite direction is set to 0. *This model must be used on faces inside the problem domain.*

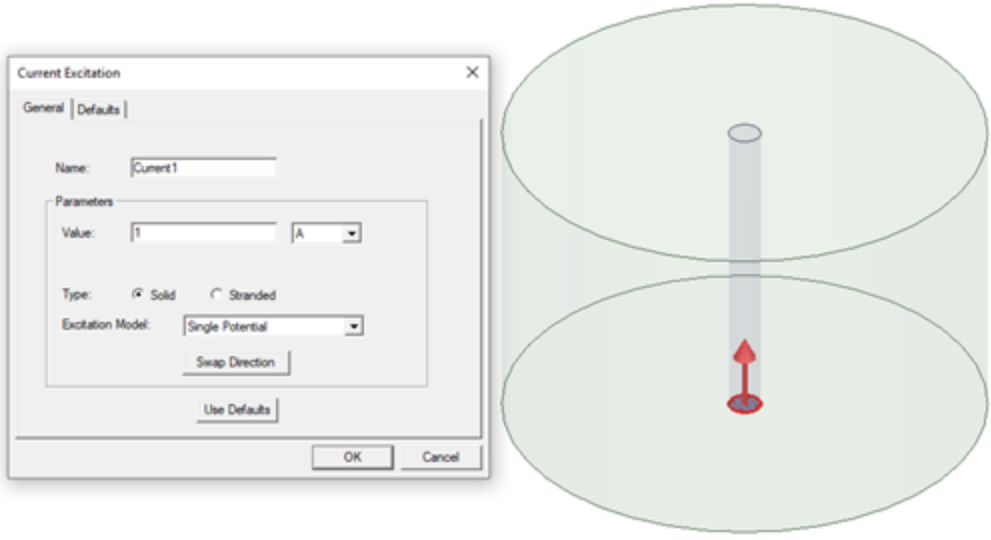




Current Excitations for A-Phi Formulation

For designs with known current values, the user selects and assigns a Current excitation on each face. The user also selects the type of conductor: solid or stranded. Current excitation supports 3 different excitation models for Solid conductor type based on how the potential DOF is assigned on the face:

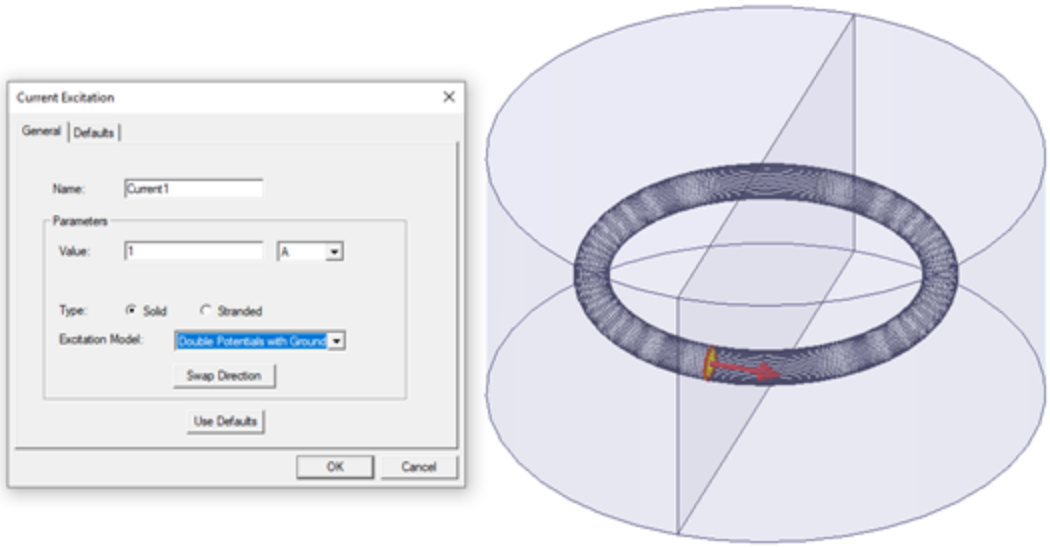
- **Single Potential**: Only one scalar electric potential DOF is assigned to the surface and it is unknown. *This model must be used only at the external boundaries of the problem domain.*

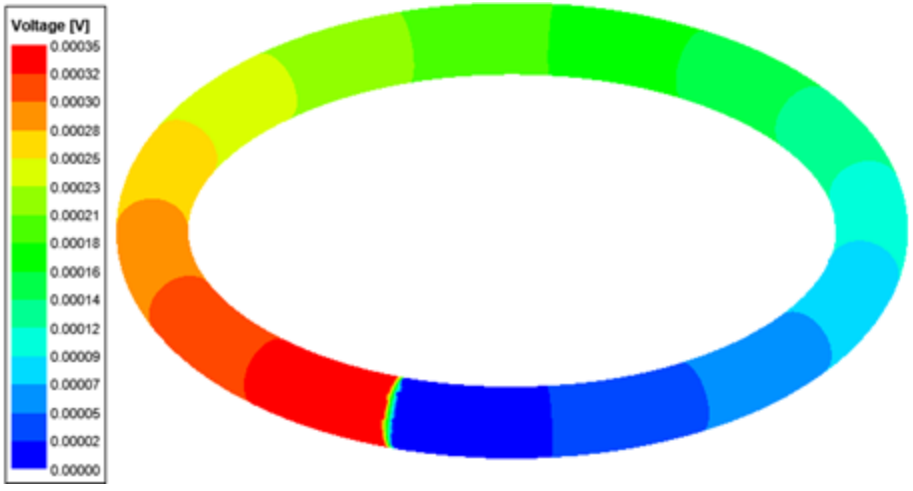


- **Double Potentials**: At the terminal face, two electric scalar potential DOFs are defined and the DOFs are unknown. There is a potential difference at the terminal. The current will be unidirectional, and it will flow in the direction that user specifies if

the current value entered is a positive quantity. *This model must be used on faces inside the problem domain.*

- **Double Potentials with Ground**: At the terminal face, two scalar electric potential DOFs are defined. The potential on the opposite side of the direction that user specified is set to 0. The current will be unidirectional, and it will flow in the specified direction if the current value entered is a positive quantity. *This model must be used on faces inside the problem domain.*





Rules/Limitations of Excitations on Conduction paths

Winding Excitations

- 1 or 2 terminals can be defined on a conduction path: if the path is a loop, then only one coil terminal per coil must be defined, else only 2 coil terminals per coil must be defined (for both solid and stranded cases).
- More than 2 coil terminals are not allowed on a conduction path.
- Coil terminals must be ordered according to the physical potential distribution. One potential degree of freedom (DOF) is assigned to each external coil terminal. Two DOFs are assigned to each internal coil terminal. For windings with external coil terminals, there are two coil terminals defined for each conduction path. The potential DOF of the second terminal on a conduction path is linked to the potential DOF of the first terminal on the next conduction path, which means the coils are serially connected. That is why it is essential to create the coil terminals in the order of the actual potential distribution on serially connected coils (conduction paths). The potential on the last Coil Terminal is set to zero. The same rule for linking the DOFs applies to the windings with internal coil terminals.

Current Excitations

- For stranded current excitation: More than 2 terminals are not allowed on a conduction path. For a loop conduction path, one terminal is defined. For other conduction paths, user must define 2 terminals (in and out). A voltage terminal and a stranded current terminal cannot be defined on the same conduction path.
- For solid current excitations: At least one voltage excitation is to be defined as a reference potential on the same conduction path of the current excitation. If a current excitation with Double Potentials with Ground selection exists in the path, or displacement current is enabled, this requirement does not apply. Multiple current excitations on a conduction path are allowed.

Voltage Excitations

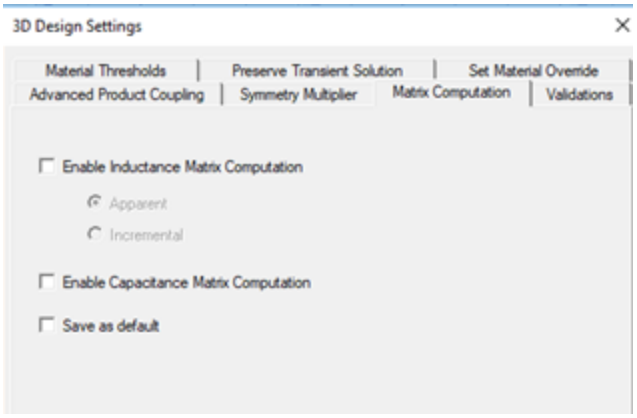
- There must be at least one voltage excitation on a conduction path with current/voltage excitations if displacement effect is not enabled (if the path is not touching the odd symmetry boundary, or there is no current/voltage excitation with Double Potentials with Ground selection, where the potential is set to 0 V).
- Winding excitation cannot be defined together with voltage or current excitations on the same path.
- On a conduction path there can be multiple solid voltage and current terminals defined.

Independent/Dependent and Odd Symmetry Boundaries

- Excitations cannot be defined on Independent/Dependent Boundaries.
- If a conduction path crosses an Odd Symmetry Boundary, Excitations must be defined at External Boundaries because the default electric potential value is zero on the Odd Symmetry Boundary. Voltage Excitation can be used to set a different potential value on the face intersecting with an Odd Symmetry Boundary. Also, any Excitation (Coil Terminal, Voltage, or Current Excitation) assignment is allowed on the face intersecting with an Odd Symmetry Boundary.
- if a conduction path crosses both Odd Symmetry and Independent/Dependent Boundaries, the User interface validation process does not allow one external terminal definition on the path. The user can select "Perform minimal validations" under Design Settings to simulate such designs at the user's own risk. Full model simulation is highly recommended for such designs.

Matrix Setup

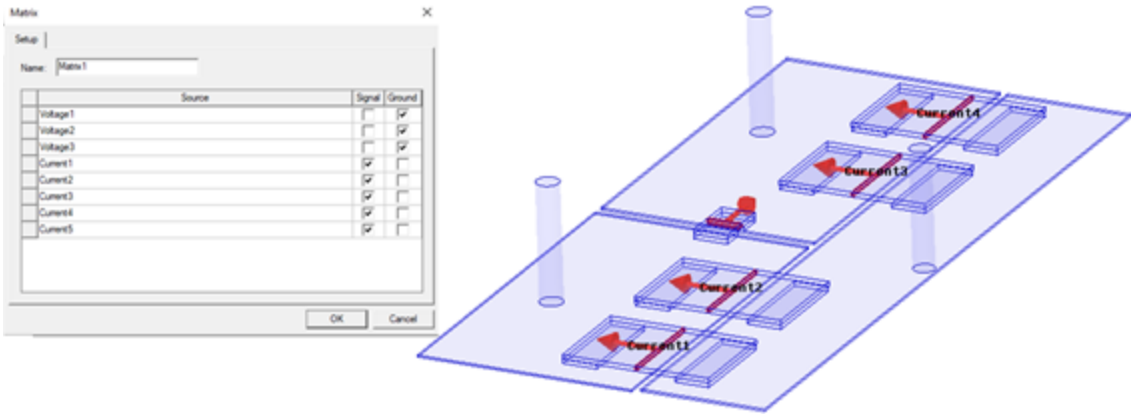
Parametric Matrix Setup is supported for the A-Phi transient solver. The user can define only one matrix setup for inductance or capacitance calculations. The user must select **Enable Inductance Matrix Computation** or **Enable Capacitance Matrix Computation** under **Design Settings** for computation of the defined matrix.



The user selects signals and grounds for the matrix of interest. For the inductance matrix, the followings are restrictions on ground selections:

- Ground selection is not allowed for windings, because they are internally grounded.
- Ground selection is not allowed for stranded current excitations since no potential is defined on the conduction path.
- Ground selection is not allowed for current excitations with double potentials and double potentials with ground.
- Ground selection is only available for Voltage excitation with Single Potential model. Ground selection is not allowed for voltage excitations with double potentials and double potentials with ground.

It is highly recommended that ground selection be as in the nominal problem setup, especially for designs with multi-terminal excitations on a conduction path. Other solver types do the matrix computation after the solution of nominal problem so it is fully post-processed, and the user can select different signal/ground/return path configurations. However, the transient solver is computing the matrix at each time step and the computation is highly dependent on how the current flows on a path, especially for designs with multiple excitations on the path.



The capacitance matrix is computed using the electrostatic energy stored in a system under 1V excitation for one selected signal and 0V for all others. Based on the selection of signals and grounds, the solver computes the capacitance matrix of the system. For the capacitance matrix, the following rule applies for the ground selection:

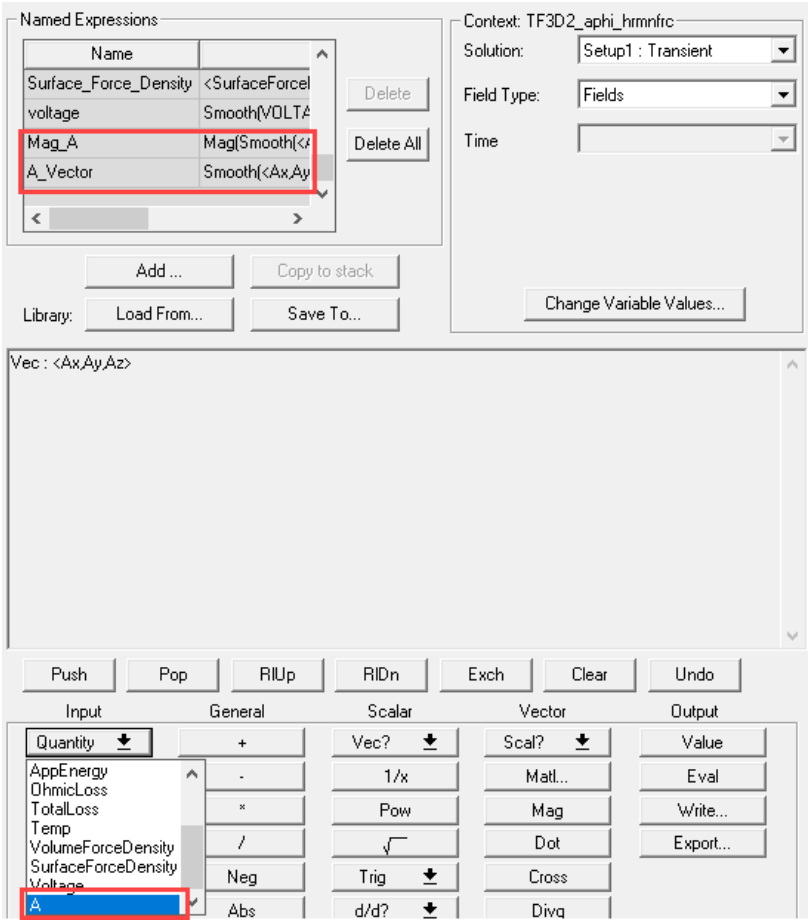
- There must be at least one ground selection in the Matrix setup if there is no internal zero reference potential defined in the design. Odd Symmetry, winding excitation, and double potentials with ground excitation model are examples of internal grounds.

Post-Processing Support

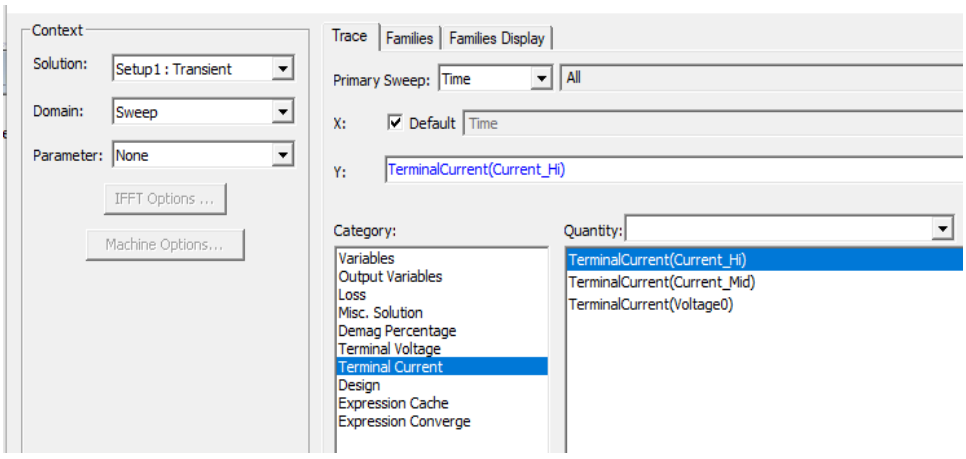
The A-phi transient solver supports all the post-processing features of the transient solver. Additionally, the following features are supported:

- Voltage in the Field Plot and in the Fields Calculator

  There are cases where no voltage is defined in an object (in a stranded object, or in an object with no eddy or displacement effect setting).
- Magnetic vector potential in the Fields Calculator



- Voltage and current quantities at the Voltage and Current Terminals
  The current computed at any terminal is assumed to be flowing into the problem domain.



Related Topics

Technical Notes: A-Phi Formulation in Maxwell 3D (Transient)

Transient A-Phi Formulation Boundaries and Excitations

Transient A-Phi Formulation Boundary Conditions

Transient A-Phi Formulation Excitations

Fields Calculator

Plotting Standard Field Quantities

Creating Reports

Selecting the Report Type

Electrostatic Solver

The electrostatic field simulator computes static electric fields in 2D or 3D due to:

- Stationary charge distributions.
- Applied potentials.

The quantity for which the electrostatic field simulator solves is the scalar electric potential, ϕ; the electric field (**E**-field) and the electric flux density (**D**-field) are automatically calculated from the potential. Derived quantities such as forces, torques, energy, surface charge density, and capacitance matrix may be calculated from these basic field quantities.

Related Topics

Technical Notes: 3D Electrostatic Field Calculation

Technical Notes: 2D Electrostatic Field Simulation

2D AC Conduction

The 2D AC conduction field simulator computes steady-state 2D electric fields in conductors due to:

- Applied potentials.

The AC conduction field solver allows you to analyze conduction currents due to time-varying electric fields in conductors and lossy dielectrics. Use it to analyze current distributions, electric field distributions and potential differences, admittances, lossy materials, and stored energy. For instance, the admittance matrix associated with a structure can be computed using the AC conduction field solver. In addition, any quantity that can be derived from the basic electromagnetic quantities can then be analyzed.

The AC conduction field solver can compute conduction currents for cartesian (XY) and axisymmetric (Cylindrical about Z) models. It assumes that all sources are sinusoids oscillating at the same frequency. Optionally, you may specify different phase angles for different sources.

You are expected to specify material properties and the electric potential at one or more object interfaces or boundaries in the model. The AC conduction field solver then computes the electric

potential, $\phi(t)$, for the model. From the electric potential, it derives the electric field, **E**(t), the electric flux density, **D**(t), and the current density, **J**(t).

Related Topics

Technical Notes: *2D AC Conduction Field Simulation*

3D AC Conduction

AC conduction analysis is the study of:

- electric field
- current
- losses
- lumped parameters (CG)

The AC conduction field solver allows you to analyze both conduction and displacement currents due to time-varying electric fields in conductors and perfect and lossy dielectrics.

Use it to analyze current distributions, electric field distributions and potential differences, and CG parameters. For instance, the admittance matrix associated with a structure can be computed using the AC conduction field solver. In addition, any quantity that can be derived from the basic electrical quantities (Electric Field, current densities, etc.) can then be analyzed.

The AC conduction field solver assumes that all sources are sinusoids oscillating at the same frequency. Optionally, you may specify different phase angles for different sources.

You need to specify these related material properties: permittivity, conductivity (for lossy materials), frequency-dependent dielectric loss tangent. You also must specify the current, electric potential at one or more object interfaces or boundaries in the model. The AC conduction field solver then computes the electric potential for the model. From the electric potential, it derives the electric field, **E**, the electric flux density, **D**, and the current density, **J**.

The supported 3D material models can be found in the table in *Maxwell Solvers Material Properties*.

Related Topics

3D AC Conduction Boundaries and Excitations

Technical Notes: 3D AC Conduction Solver Theory

Maxwell Solvers Material Properties

DC Conduction Solver

The electric conduction field simulator computes steady-state 2D or 3D electric fields in conductors due to:

- Current excitations.
- Applied potentials.

The quantity for which the electric conduction field simulator solves is the electric potential, ϕ; the electric field (**E**-field**)** and the current density (**J**-field) are automatically calculated from the potential. The resistance matrix, a derived quantity, may be calculated from these basic field quantities. As an additional option, perfect insulators, that is, non-conduction regions, surrounding the conductors can also be added to the simulation domain allowing to calculate the electric field everywhere including the insulators.

Related Topics

DC Conduction Boundaries and Excitations

Technical Notes: 2D DC Conduction Field Simulation

Electric Transient Solver

The 3D electric transient field simulator computes time-varying electric fields excited by:

- Time-varying applied potentials.
- Total charge and volume charge density.
- Time-varying current excitations.

The quantity for which the electric transient field simulator solves is the electric potential, ϕ by solving the following equation:

$$-\nabla \cdot \left(\varepsilon \nabla \frac{\partial \Phi}{\partial t}\right) - \nabla \cdot (\sigma \nabla \Phi) = 0$$

The electric field (**E**-field), the current density (**J**-field), and the electric flux density (**D**-field) are automatically calculated from the potential.

Derived solution parameters such as the electric energy, Ohmic loss, surface charge density, and maximum electric field can be obtained for each solid.

The temporal discretization is adaptive and, besides the initial and maximum time step, it also depends on temporal tolerance. The default temporal tolerance is 0.005, but it depends on the problem to be solved.

Normally, problems with high time derivative field solutions would require smaller temporal tolerances.

In the case of non-linear problems, temporal tolerance also depends on the non-linearity of materials and its value needs to be set around 1e-12 or even smaller.

Maxwell Solvers Material Properties

The tables below list material properties supported by the Maxwell 2D and 3D solvers. Additional information may be found in the sections on *Adding Materials*, *Assigning Material Property Types*, and *Materials*.

Material Properties for Maxwell 3D Solvers

Material property	Electrostatic	DC Conduction	AC Conduction	Electric Transient	Magnetostatic	Eddy current	Magnetic Transient (T-Omega)	Magnetic Transient (A-Phi)
Permittivity								
Linear (Simple)	X	X	X	X		X		X
Non-linear	X	X	X					
Isotropic	X	X	X	X		X		X
Anisotropic	X	X	X	X		X		X
Non-linear anisotropy								
Frequency dependent			X			X		
Conductivity								
Linear (Simple)		X	X	X	X	X	X	X
Non-linear		X	X	X				
Isotropic		X	X	X	X	X	X	X
Anisotropic		X	X	X	X	X	X	X
Non-linear anisotropy								
Frequency dependent			X			X		
Dieletric loss tangent								
Linear (Simple)			X			X		
Non-linear								
Isotropic			X			X		
Anisotropic			X					
Non-linear anisotropy								
Frequency dependent			X			X		
Permeability								
Linear (Simple)					X	X	X	X
Non-linear					X	X (1)	X	X
Isotropic					X	X	X	X

Material property	**Electrostatic**	**DC Conduction**	**AC Conduction**	**Electric Transient**	**Magnetostatic**	**Eddy current**	**Magnetic Transient (T-Omega)**	**Magnetic Transient (A-Phi)**
Anisotropic					X	X	X	X
Non-linear anisotropy					X	X	X	X
Vectorial hysteresis							X	
Frequency dependent						X		
Magnetic loss tangent								
Linear (Simple)						X		
Non-linear								
Isotropic						X		
Anisotropic						X		
Non-linear anisotropy								
Frequency dependent						X		
Composition								
Solid					X	X	X	X
Lamination					X (2)	X (2)	X (2)	X
Litz Wire					X (3)	X (3)	X (3)	X
Magnetostriction					X(4)		X(4)	
Notes:								
(1) This specific non-linear model is different from other solvers								

Material property	Electrostatic	DC Conduction	AC Conduction	Electric Transient	Magnetostatic	Eddy current	Magnetic Transient (T-Omega)	Magnetic Transient (A-Phi)
(2-3)Lamination and Litz wire have different models in the magnetostatic, eddy current and magnetic transient solvers								
(4)Magnetostriction has different models in the magnetostatic and magnetic transient solvers								

[Material Properties for Maxwell 2D Solvers]

Material property	Electrostatic	DC Conduction	AC Conduction	Magnetostatic	Eddy current	Magnetic Transient
Permittivity						
Linear (Simple)	X		X		X	
Non-linear						
Isotropic	X		X		X	
Anisotropic	X		X		X	
Non-linear anisotropy						
Frequency dependent			X		X	
Conductivity						
Linear (Simple)	X	X	X	X	X	X
Non-linear						X
Isotropic	X	X	X		X	X
Anisotropic		X	X		X	X

Material property	Electrostatic	DC Conduction	AC Conduction	Magnetostatic	Eddy current	Magnetic Transient
Non-linear anisotropy						
Frequency dependent			X		X	
Dieletric loss tangent						
Linear (Simple)			X		X	
Non-linear						
Isotropic			X		X	
Anisotropic					X	
Non-linear anisotropy						
Frequency dependent			X		X	
Permeability						
Linear (Simple)				X	X	X
Non-linear				X	X (1)	X
Isotropic				X	X	X
Anisotropic				X	X	X (5)
Non-linear anisotropy				X		X(5)
Vectorial hysteresis						X
Frequency dependent					X	
Magnetic loss tangent						
Linear (Simple)					X	
Non-linear						
Isotropic					X	
Anisotropic					X	
Non-linear anisotropy						
Frequency dependent					X	

Material property	Electrostatic	DC Conduction	AC Conduction	Magnetostatic	Eddy current	Magnetic Transient
Composition						
Solid				X	X	X
Lamination				X	X (2)	X (2)
Litz Wire					X (3)	X (3)
Magnetostriction				X(4)		X(4)
Notes:						
(1) This specific non-linear model is different from other solvers						
(2-3)Lamination and Litz wire have different models in the magnetostatic, eddy current and magnetic transient solvers						
(4) Magnetostriction has different models in the magnetostatic and magnetic transient solvers						
(5) Supported only in 2D transient designs with Cartesian XY symmetry						

Related Topics

Specifying the Solver Type

Disconnected Solution Domains

There are two ways to create a problem that contains multiple disconnected solution domains:

- By not creating or by excluding (not solving in) the background object, when the included objects are not all touching.
- By using excluded objects or perfect conductors that cut the included objects into different regions.

The Maxwell solver identifies multiple disconnected solution domains.

For the electric solvers (electrostatic, conduction, and conduction+electrostatic), the solution is allowed even if multiple disconnected solution domains are detected. In this case, an error message appears only if a solution domain with inconsistent or unsolvable solution setup is detected.

For the magnetic solvers (magnetostatic, eddy current, and transient EM), an error message appears when multiple disconnected solution domains are detected.

8 - Drawing a Model

After you insert a design into the current project, you can draw a model of the electromagnetic structure. The general strategy is to build the model as a collection of 3D objects. You can assign any single material to each 3D object.

You can create 3D objects by using the modeler's **Draw** commands or you can draw 1D and 2D objects, and then manipulate them to create 3D objects. Objects are drawn in the **3D Modeler** window. You can also import objects from other systems.

To open a new **3D Modeler** window, do one of the following:

- Insert a new design into the current project.
- Double-click a design in the project tree.

If a **3D Modeler** window is not open, do one of the following:

- On the Maxwell menu, click **3D Model Editor**.
- Right-click the design name in the project tree, and then click **3D Model Editor** on the shortcut menu.

The model you draw is saved with the current project when you click **File>Save**.

Note	If you access your machine via Remote Desktop, if Maxwell is running and one or more modeler windows are open, those modeler windows automatically close. The message manager window displays a message indicating that Maxwell closed the modeler windows.

When working with multiple projects, or when a project has multiple designs, you may have multiple **Modeler** windows available. To switch to the modeler window associated with a specific design:

1. In the Project Manager window, select the **Design** of interest.
2. Click **Maxwell3D Model Editor** to focus the modeling window on the selected design.

Note	If the menu command is unavailable, then the selected design is already in the modeler window.

Related Topics

Drawing Objects

Model Analysis

Design Settings

Setting the Temperature of Objects

Creating a 3D Component from an Existing Model

3D Component Library

Creating a User Define Primitive

Modifying Objects

Selecting Objects

Choosing the Movement Mode

Drawing Objects

You can draw one-, two-, or three-dimensional objects using the **Draw** commands. You can alter objects individually or together to create the geometry of your structure. In the **Tools>Modeler Options, Drawing** tab, you can set a default to either draw objects directly with the mouse or by invoking a **Properties** dialog in which you can enter the values for the object dimensions. The **Dialog** mode drawing feature works with the equation based line, and all two and three dimensional objects. You can toggle to **Point** mode via the **F3** function key and to **Dialog** mode via the **F4** function key. When you use the **Dialog** mode for drawing objects the **Edit property of new primitives** setting is ignored.

One-dimensional (1D) objects in the modeler include straight line, arc line, center-point arc, and spline segments, or a combination of these - called polylines. One-dimensional objects are open objects; their boundaries do not enclose a region, unless you connect their endpoints. They have length, but no surface or volume. Generally they are used as temporary objects from which to create 2D objects.

Two-dimensional (2D) objects in the modeler include objects such as equation based surfaces, rectangles, ellipses, circles, and regular polygons. Two-dimensional objects are closed sheet objects; their boundaries enclose a region. You can create 2D sheet objects by covering the enclosed region. In many applications (FSS, antennas) it is essential to calculate net power flow through a surface.

You can also edit the properties of a polyline from the history tree to assign it a Cross Section property as line or rectangular. If you then assign it either a height or a width, the polyline becomes a sheet object.

By default, the history tree organizes sheet objects according to their boundary assignments. To change this, select the **Sheets** icon, and right-click to display the **Group Sheets by Assignment** check box. Within the calculator sheet objects are listed under surface.

Three-dimensional (3D) objects in the modeler include objects such as boxes, cylinders, regular polyhedra, cones, spheres, torii, and helices. These objects have boundaries that enclose a region with volume.

You can create 3D objects by manipulating 2D objects along a plane or by using the appropriate **Draw** commands. You can also edit the properties of a polyline from the history tree to assign it a Cross Section property as circle rectangular. If you then assign it an appropriate diameter or both height or a width, the polyline becomes a 3D object.

By default, the history tree organizes 3D objects by material. To change this, select the **Objects** icon, and right click to display the **History Tree Layout Commands**. You can also use Group Commands for Modeler Objects.

While you draw objects you can also:

- Select Movement Mode as 3D, In Plane, Out of Plane, Along X, Y or Z axis.
- Select Grid Plane as XY, YZ, or XZ.
- Set Snap Mode
- Set Reference Point for the movement mode
- Adjust the View

After you draw an object in the **3D Modeler** window, you can modify the object's properties, such as its position, dimensions, or color, in the **Properties** dialog box. Most object properties can be assigned to variables that can then be manipulated during the solve to test their effect on the solution.

Note	If you access your machine via Remote Desktop, if Maxwell is running and one or more modeler windows are open, those modeler windows automatically close. The message manager window displays a message indicating that Maxwell closed the modeler windows.

Related Topics

Assigning a Cross Section and Dimension to a Polyline

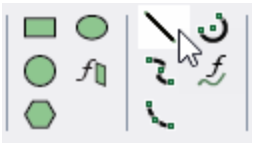
Modifying Objects

Drawing a Region

Setting the Temperature of Objects

Drawing a Straight Line Segment

To create an object with one or more straight line segments, use the **Draw>Line** command.

1. Click **Draw>Line**  .
2. Select the first point of the line in one of the following ways:
   - Click the point.
   - Edit the point's coordinates in the **X**, **Y**, and **Z** boxes in the status bar.

   To delete the last point that was entered, click **Undo Previous Segment** on the shortcut menu. After using the undo feature, you can also use **Redo Previous Segment** on the shortcut menu.
3. Select the endpoint of the line by clicking the point or typing the coordinates in the text boxes in the status bar.

   The endpoint serves as the start point for a subsequent line segment.

   To delete all points and start over, press **ESC** or click **Escape Draw Mode** on the shortcut menu.
4. Complete the line in one of the following ways:

- Double-click the endpoint.
- Click **Done** on the shortcut menu.
- Press **Enter**.

If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes. Those listed under the Command tab describe the commands used to create the object. These commands also appear in the History tree. The Properties listed as line attributes include Name, Orientation, whether a Model object, whether to Display Wireframe, Color, Transparency, and whether to Show Direction as arrows. The Show Direction property is most helpful to unambiguously show the line start orientation when plotting fields along a line.

5. Click **OK to close the Properties dialog.**

Note	While drawing a polyline, you can switch between straight line, arc line, or spline segments using the **Set Edge Type** commands on the shortcut menu.

Related Topics

Setting the Reference Point
Deleting Polyline Segments
Converting Polyline Segments
Drawing a Three-Point Arc
Drawing a Center-Point Arc Line
Drawing a Spline
Drawing a Polyline

Drawing a Three-Point Arc Line

In the modeler, a three-point arc line segment is an arced line defined by three points on its curve. Use the **Draw>Arc>3 Point** command to create a polyline object with one or more arc line segments.

1. Click **Draw>Arc>3 Point**  .
2. Select the start point of the arc in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the **X**, **Y**, and **Z** text boxes in the status bar.
3. Select the midpoint of the arc by clicking the point or typing the coordinates in the **X**, **Y**, and **Z** text boxes in the status bar.

   To delete the last point that was entered, click **Undo Previous Segment** on the shortcut menu. After using the undo feature, you can also use **Redo Previous Segment** on the shortcut menu.

   To delete all points and start over, press **ESC** or click **Escape Draw Mode** on the shortcut menu.

4. Select the endpoint of the arc by clicking the point or typing the coordinates in the **X**, **Y**, and **Z** text boxes in the status bar.

   The endpoint serves as the start point for a subsequent arc line segment.

5. If the endpoint is the last point of the polyline object, double-click the point to complete the polyline or click **Done** on the shortcut menu.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes. Those listed under the Command tab describe the commands used to create the object. These commands also appear in the History tree. The Properties listed as line attributes include Name, Orientation, whether a Model object, whether to Display Wireframe, Color, Transparency, and whether to Show Direction as arrows. The Show Direction property is most helpful to unambiguously show the line start orientation when plotting fields along a line.

6. Click **OK**.

   Based on the three points you specified, the modeler calculates the center point and radius of the arc and draws an arced line through the three points.

Note	While drawing a polyline, you can switch between arc line, straight line, or spline segments using the **Set Edge Type** commands on the shortcut menu.

Related Topics

Setting the Reference Point
Drawing a Straight Line
Drawing a Spline
Drawing a Center-Point Arc Line
Drawing a Polyline
Deleting Polyline Segments
Converting Polyline Segments

Drawing a Center-Point Arc Line

In the modeler, a center-point arc line segment is an arced line defined by a center point, start point and angle. Use the **Draw>Arc>Center Point** command to create a polyline object with one or more center-point arc line segments.

1. Click **Draw**>**Arc**>**Center Point**  .
2. Select the center point of the arc in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the **X**, **Y**, and **Z** text boxes in the status bar.
3. Select the start point, or radius, of the arc by clicking the point or typing the coordinates in the **X**, **Y**, and **Z** text boxes in the status bar.

To delete the last point that was entered, click **Undo Previous Segment** on the shortcut menu. After using the undo feature, you can also use **Redo Previous Segment** on the shortcut menu.

To delete all points and start over, press **ESC** or click **Escape Draw Mode** on the shortcut menu.

4. Sweep the angle, or endpoint, of the arc by clicking the point or typing the coordinates in the **X**, **Y**, and **Z** text boxes in the status bar.
5. If the endpoint is the last point of the polyline object, double-click the point to complete the polyline or click **Done** on the shortcut menu.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes.
6. Click **OK**.

Note	While drawing a polyline, you can switch between arc line, straight line, or spline segments using the **Set Edge Type** commands on the shortcut menu.

Related Topics

Setting the Reference Point
Drawing a Straight Line
Drawing a Spline
Drawing a Polyline
Deleting Polyline Segments
Converting Polyline Segments
Drawing a Three-Point Arc

Drawing a Spline

A spline is a curved line defined by three points. The modeler uses a natural spline type: a piece wise cubic spline with an end condition that has a derivative of zero. Use the **Draw>Spline** command to create a polyline object with one or more spline segments.

1. Click **Draw**>**Spline**  .
2. Select the spline's start point in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the **X**, **Y**, and **Z** text boxes in the status bar, and then press **Enter**.

   To delete the last point entered, click **Undo Previous Segment** on the shortcut menu.After using the undo feature, you can also use **Redo Previous Segment** on the shortcut menu.

   To delete all selected points and start over, press **ESC** or click **Escape Draw Mode** on the shortcut menu.

3. Select the midpoint of the spline by clicking the point or typing the coordinates in the **X**, **Y**, and **Z** text boxes in the status bar.
4. Select the endpoint of the spline by clicking the point or typing the coordinates in the **X**, **Y**, and **Z** text boxes in the status bar.

   The endpoint serves as the start point for a subsequent spline segment.
5. Complete the spline in one of the following ways:
   - Double-click the endpoint.
   - Click **Done** on the shortcut menu.
   - Press **Enter**.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes. Those listed under the Command tab describe the commands used to create the object. These commands also appear in the History tree. The Properties listed as line attributes include Name, Orientation, whether a Model object, whether to Display Wireframe, Color, Transparency, and whether to Show Direction as arrows. The Show Direction property is most helpful to unambiguously show the line start orientation when plotting fields along a line.
6. Click **OK**.

Note	While drawing a polyline, you can switch between spline, straight line, or arc line segments using the **Set Edge Type** commands on the shortcut menu.

Related Topics

Setting the Reference Point

Drawing a Polyline

Drawing a Center-Point Arc Line

Deleting Polyline Segments

Converting Polyline Segments

Drawing a Polyline

A polyline is a single object that includes any combination of straight line, arc line, or spline segments. The endpoint of one segment is the start point for the next segment. Use the shortcut menu's **Set Edge Type** commands to switch between straight line, arc line, or spline segments while drawing a polyline.

In the **Polyline** section of **Operation** tab of the **Modeler Options**, select or clear the **Automatically cover closed polylines** check box.

If checked, closed polylines become sheet objects, and are listed as such in the History tree. If unchecked, closed polylines are listed under lines in the History tree.

1. Click **Draw**>**Line**  .
2. Right-click in the **3D Modeler** window to access the shortcut menu, and then point to **Set Edge Type**.
3. Click **Straight**, **Spline**, **3 Point Arc**, or **Center Point Arc** depending on which type of polyline segment you want to draw.
4. If you clicked **Straight**, follow the procedure for drawing a straight line.

   If you clicked **Spline**, follow the procedure for drawing a spline.

   If you clicked **3 Point Arc**, follow the procedure for drawing a three-point arc line.

   If you clicked **Center Point Arc**, follow the procedure for drawing a center-point arc line.
5. Repeat steps 2 and 3 for each segment of the polyline object. The endpoint of the previous segment serves as the start point for the next segment.

   The shortcut menu lets you do the following for each segment:

   Undo Previous Segment or **Redo Previous Segment**.
6. Complete the polyline in one of the following ways:
   - Double-click the endpoint of the final segment.
   - Click **Done** on the shortcut menu.

Note	To connect the polyline's start and endpoints, click **Close Polyline** on the shortcut menu.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes. Those listed under the Command tab describe the commands used to create the object. These commands also appear in the History tree. The Properties listed as line attributes include Name, Orientation, whether a Model object, whether to Display Wireframe, Color, Transparency, and whether to Show Direction as arrows. The Show Direction property is most helpful to unambiguously show the line start orientation when plotting fields along a line. Notice that by going to the History tree selecting Create Polyline for that object, you can assign a cross-section and dimensions to a polyline.
7. Click **OK**.

Related Topics

Setting the Reference Point

Assigning a Cross Section and Dimension to a Polyline

Drawing a Straight Line

Drawing a Three-Point Arc

Drawing a Center-Point Arc Line

Drawing an Equation-Based Curve

Drawing a Spline

Deleting Polyline Segments

Converting Polyline Segments

Generate History

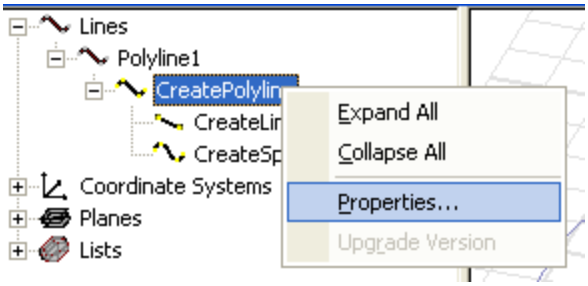
Setting Modeler Options: Operations Tab

Assigning a Cross Section and Dimensions to a Polyline

By viewing the History tree property of a polyline, you can assign either a line, circle or rectangle or isosceles trapezoid cross-section to a polyline. This assignment enables editable dimension properties of width for a line, diameter for a circle, and height and width for a rectangle. To assign a cross-section to a polyline:

1. In the History tree of the Modeler window, right-click on the polyline that you want to give a cross-section.

   This selects the polyline, displays the polyline properties in the docked properties (if you have if displays) and displays a shortcut menu where you can choose **Properties...** to display the undocked Properties window for the polyline.



2. In a Properties window (either docked or undocked) for the selected polyline click on **None** on the **Type** line under Cross Section to display the choices for Line, Circle, Rectangle or Isosceles Trapezoid.
3. Select one of Line, Circle, Rectangle, or Isosceles Trapezoid as the cross-section.
   - Selecting **Line** causes the Cross Section area of the polyline properties to display editable fields for Orientation and Width.
   - Selecting **Circle** causes the Cross Section area of the polyline properties to display an editable field for diameter.
   - Selecting **Rectangle** causes the Cross Section area of the polyline properties to display editable fields for Orientation, Width and Height.
   - Selecting **Isosceles Trapezoid** causes the Cross Section area of the polyline properties to display editable fields for Orientation, Width/Diameter, Top Width and Height.
4. If you select Line, Rectangle or Isosceles Trapezoid, you can edit the Orientation as Auto, X, Y, or Z. This provides the direction in which the dimension extends.
5. Specify the dimensions and select the units for the Cross section.

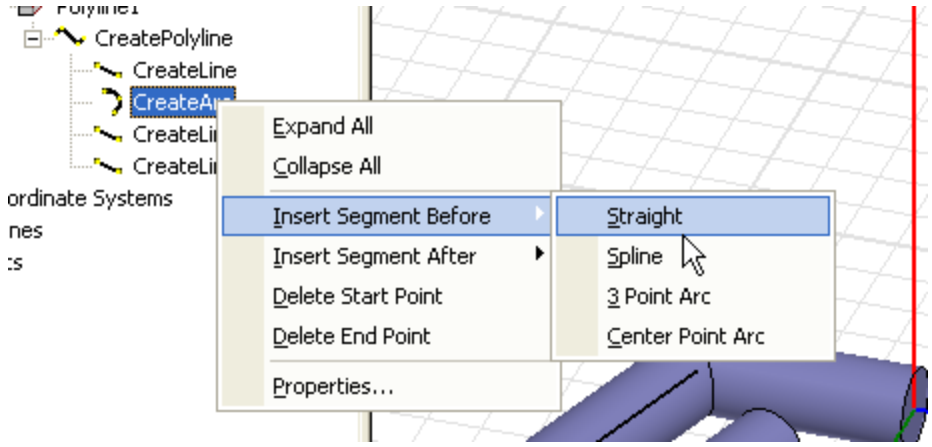
   Type a value in the dimension field(s) and select units from the drop down menu.

   The dimensions must be reasonable relative to the specified shape and orientation of the polyline. If the polyline cannot be extended into current Orientation for the given dimension

(s), you will receive a warning. If you receive a warning, check the Orientation, dimension and units.

When the modeler can extend the dimensions legally, it displays the modified object, and lists it in the History tree as either a Sheet object (Line or one dimensional Rectangle) or as a Solid object (Circle or two dimensional Rectangle).

6. You can modify the new polyline either by editing the properties, or by using the History tree to select one of the line objects that make up the polyline, and right clicking to display the popup menu showing commands to **Insert**, **Delete**, or display editable segment **Properties**.



A segment that you select in the Project tree is indicated in the Modeler window by a line in the dimensioned object. If you insert a new segment, it adopts the dimensions you specified for the polyline object.

Related Topics

Setting the Reference Point
Drawing a Polyline
Drawing an Equation-Based Curve
Deleting Polyline Segments
Inserting Line Segments
Drawing a Center-Point Arc Line
Drawing a Straight Line
Drawing a Three-Point Arc

Inserting Line Segments

You can insert line segments of various kinds for existing line objects.

1. Select the line object in the **History** tree (not the modeler window).

This highlights the object and enables the **InsertLine Segment** commands in the **Draw** menu and short-cut menu.

2. Use the cascade menu from the **Draw>Line Segment** command to or the right-click menu to select whether to **Insert Before Line Segment** or **InsertAfter Line Segment**.
3. Use the next cascade menu to specify the kind of segment to add. These can be: Straight, Spline. 3 Point Arc, or Center Point Arc.
4. If you clicked **Straight**, follow the procedure for drawing a straight line.

   If you clicked **Spline**, follow the procedure for drawing a spline.

   If you clicked **3 Point Arc**, follow the procedure for drawing a three-point arc line.

   If you clicked **Center Point Arc**, follow the procedure for drawing a center-point arc line.
5. Repeat steps 2 and 3 for each segment of the polyline object. The endpoint of the previous segment serves as the start point for the next segment.
6. Complete the polyline in one of the following ways:
   - Double-click the endpoint of the final segment.
   - Click **Done** on the shortcut menu.

Note	To connect the polyline's start and endpoints, click **Close Polyline** on the shortcut menu.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes. Those listed under the Command tab describe the commands used to create the object. These commands also appear in the History tree. The Properties listed as line attributes include Name, Orientation, whether a Model object, whether to Display Wireframe, Color, Transparency, and whether to Show Direction as arrows. The Show Direction property is most helpful to unambiguously show the line start orientation when plotting fields along a line.
7. Click **OK**.

Related Topics

Drawing a Center-Point Arc Line
Deleting Polyline Segments
Converting Polyline Segments

Drawing an Equation-Based Curve

Any line that can be described by an equation in three dimensions can be drawn.

1. Click **Draw**>**Equation Based Curve** .

   The **Equation Based Curve** dialog box opens.

2. Type equations for **X(_t)**, **Y(_t)**, and **Z(_t)**.

Note	You can also define an equation by doing the following: 1. Click the **...** button. The **Edit Equation** dialog box appears. 2. Do one or more of the following to define the equation: • Type a numerical value or expression directly in the text box. • Select a function to insert from the pull-down list, and select **Insert Function**. • Select an operator from the pull-down list, and select **Insert Operator**. • Select a quantity from the pull-down list, and select **Insert Quantity**. 3. When you are finished defining the equation, click **OK** to close the **Edit Equation** dialog box and return to the **Equation Based Curve** dialog box.

   - Any unitless value input in equation based curve is taken as model units. For example, for Y(_t) = 1, the y value is taken as 1 model units (say mm). If a value has units, then it is converted to model units and used. For example, if we specify Y(_t) = 1cm, then y value will be correctly taken as 10mm.
   - While parsing expressions, equation based curves convert each variable separately to model units and assume that the resulting expression is in model units.
   - Equation based curves depend on the variable value library to correctly evaluate the units of expression.
3. Select a start value from the **Start_t** pull-down list.
4. Select an end value from the **End_t** pull-down list.
5. Select the number of points in the curve from the **Points** pull-down list.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes. Those listed under the Command tab describe the commands used to create the object. These commands also appear in the History tree. The Properties listed as line attributes include Name, Orientation, whether a Model object, whether to Display Wireframe, Color, Transparency, and whether to Show Direction as arrows. The Show Direction property is most helpful to unambiguously show the line start orientation when plotting fields along a line.
6. Click **OK** on the **Properties** dialog box.

Related Topics

Setting the Reference Point

Assigning a Cross Section and Dimension to a Polyline

Drawing a Circle

Draw a circle by selecting a center point and a radius. Circles are drawn as true surfaces in the modeler.

1. On the **Draw** menu, click **Circle**  .
2. Select the center point of the circle in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Specify the radius by selecting a point on the circle's circumference in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the center point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes.
4. Click **OK**.

   If the **Automatically cover closed polyline** option is selected in the **Modeler Options** window, the circle will be covered, resulting in a 2D sheet object. Otherwise it will be a closed 1D polyline object.

Note	The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see: Technical Notes, "Surface Approximations" and related sections, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Related Topics

Setting the Reference Point
Surface Approximation
Creating Segmented Geometry
Covering Lines

Drawing an Ellipse

Draw an ellipse by specifying a center point, base radius, and secondary radius.

1. On the **Draw** menu, click **Ellipse**  .
2. Select the center point of the ellipse in one of the following ways:

- Click the point.
- Type the point's coordinates in the text boxes in the status bar.

3. Specify the base radius of the ellipse. If the current drawing plane is xy, then x is the base radius direction. If the drawing plane is yz, then y is the base radius direction. If the drawing plane is xz, then z is the base radius direction. Select the point in one of the following ways:
   - Click the point. Maxwell constrains mouse movement to the base radius direction.
   - Type the coordinates of a point relative to the center point in the **dX**, **dY**, or **dZ** box, where **d** is the distance from the previously selected point.
4. Specify the secondary radius of the ellipse. Select the point in one of the following ways:
   - Click the point. Maxwell constrains mouse movement to a point on the plane orthogonal to the base radius direction.
   - Type the coordinates of a point relative to the center point in the **dX**, **dY**, or **dZ** box.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's properties.

   The **Ratio** value represents the aspect ratio of the secondary radius to the base radius.
5. Click **OK**.

   If the **Automatically cover closed polyline** option is selected in the **Modeler Options** window, the ellipse will be covered, resulting in a 2D sheet object. Otherwise it will be a closed 1D polyline object.

If the base radius is larger than the secondary radius, the ellipse's longer axis will lie along the default base radius direction. If the secondary radius is larger than the base radius, the ellipse's longer axis will lie perpendicular to the default base radius direction. To create an ellipse with an arbitrary orientation, rotate or move the ellipse after drawing it.

Note	The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see: Technical Notes, "Surface Approximations" and related sections, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Related Topics

Setting the Reference Point

Modifying Surface Approximation Settings

Creating Segmented Geometry

Covering Lines

Drawing a Rectangle

Draw a rectangle (or square) by selecting two diagonally opposite corners.

1. Click **Draw**>**Rectangle**  .
2. Select the first diagonal corner in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.

   To delete the selected point and start over, press **ESC** or click **Escape Draw Mode** on the shortcut menu.
3. Select the second corner of the rectangle in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the first diagonal corner in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's properties.
4. Click **OK**.

   If the **Automatically cover closed polyline** option is selected in the **Modeler Options** window, the rectangle will be covered, resulting in a 2D sheet object. Otherwise it will be a closed 1D polyline object.

Related Topics

Setting the Reference Point

Covering Lines

Drawing a Regular Polygon

A regular polygon is a 2D object with three or more equal sides. Regular polygons are useful for drawing faceted 2D objects.

1. Click **Draw**>**Regular Polygon**  .
2. Select the center point of the polygon in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Specify the polygon's radius, the distance from the center point to one of the polygon's vertices, in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the center point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

4. In the **Segment number** dialog box, enter the **Number of segments** in the polygon, and then click **OK**.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's properties.

5. Click **OK**.

Note	The radius is measured from the center point to a corner of the polygon, or the intersection of two edges. It is *not* measured from the center point to the midpoint of an edge.

   If the **Automatically cover closed polyline** option is selected in the **Modeler Options** window, the polygon will be covered, resulting in a 2D sheet object. Otherwise it will be a closed 1D polyline object.

Related Topics

Setting the Reference Point

Covering Lines

Drawing an Equation-Based Surface

Any surface that can be described by an equation in three dimensions can be drawn.

1. Click **Draw>Equation Based Surface** .

   The **Equation Based Surface** dialog box opens.

2. Type equations for **X(_u, _v)**, **Y(_u, _v)**, and **Z(_u, _v)**.

Note	You can also define an equation by doing the following: 1. Click the **...** button. The **Edit Equation** dialog box appears. 2. Do one or more of the following to define the equation: • Type a numerical value or expression directly in the text box. • Select a function to insert from the pull-down list, and select **Insert Function**. • Select an operator from the pull-down list, and select **Insert Operator**. • Select a quantity from the pull-down list, and select **Insert Quantity**. 3. When you are finished defining the equation, click **OK** to close the **Edit Equation** dialog box and return to the **Equation Based Surface** dialog box.

- Any unitless value input in equation based curve is taken as model units. For example, for Y(_t) = 1, the y value is taken as 1 model units (say mm). If a value has units, then it is converted to model units and used. For example, if we specify Y(_t) = 1cm, then y value will be correctly taken as 10mm.
- While parsing expressions, equation based curves convert each variable separately to model units and assume that the resulting expression is in model units.

- Equation based curves depend on the variable value library to correctly evaluate the units of expression.

3. Select start values from the **Start_u** and **Start_v** pull-down lists.

   Select end values from the **End_u** and **End_v** pull-down lists.If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's properties.

4. Click **OK** on the **Properties** dialog box.

Related Topics

Setting the Reference Point

Assigning a Cross Section and Dimension to a Polyline

Drawing an Equation Based Curve

Drawing a Sphere

Draw a sphere by selecting a center point and a radius. Spheres are drawn as true surfaces in the modeler.

1. Click **Draw**>**Sphere**  .
2. Select the center point of the sphere in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Specify the radius by selecting a point on the sphere's circumference in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the center point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's properties.
4. Click **OK**.

Note	The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see: Technical Notes, "Surface Approximations" and related sections, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Drawing a Cylinder

Draw a cylinder by selecting a center point, radius, and height. Cylinders are drawn as true surfaces in the modeler.

1. Click **Draw>Cylinder**  .
2. Select the center point of the cylinder's base circle in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Specify the radius by selecting a point on the base circle's circumference in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the center point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.
4. Specify the cylinder's height by selecting a point on the axis perpendicular to the base circle's plane. Select the point by clicking the point or typing the coordinates in the **dX**, **dY**, and **dZ** boxes.

Note	If you create a cylinder with a height of zero, Maxwell draws a circular sheet object.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's attributes.
5. Click **OK**.

Note	The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see: Technical Notes, "Surface Approximations" and related sections, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Related Topics

Setting the Reference Point

Assigning a Cross Section and Dimension to a Polyline

Modifying Surface Approximation Settings

Creating Segmented Geometry

Drawing a Box

Draw a box by selecting two diagonally opposite corners of the base rectangle, then specifying the height.

1. Click **Draw**>**Box**  .
2. Select the first diagonal corner of the base rectangle in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.

   To delete the selected point and start over, press **ESC** or click **Escape Draw Mode** on the shortcut menu.
3. Select the second corner of the base rectangle in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the first diagonal corner in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.
4. Specify the height of the box by selecting a point on the axis perpendicular to the base rectangle. Select the point by clicking the point or typing the coordinates in the **dX**, **dY**, and **dZ** boxes.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's properties.
5. Click **OK**.

Related Topics

Setting the Reference Point

Drawing a Regular Polyhedron

In the modeler, regular polyhedrons are 3D objects with regular polygon faces; each face has three or more equal sides. Regular polyhedrons are useful for drawing faceted 3D objects.

1. Click **Draw**>**Regular Polyhedron**  .
2. Select the center point of the polyhedron in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Select the radius of the polyhedron, the distance from the center point to one of the polyhedron's vertices, in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the center point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

4. In the **Segment number** dialog box, enter the **Number of segments** in the polyhedron, and then click **OK**.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's properties.

5. Click **OK**.

Note	The radius is measured from the center point to a corner of the polygon, or the intersection of two edges. It is *not* measured from the center point to the midpoint of an edge.

   If the **Automatically cover closed polyline** option is selected in the **Modeler Options** dialog box, the polygon will be covered, resulting in a 2D sheet object. Otherwise it will be a closed 1D polyline object.

Related Topics

Setting the Reference Point

Covering Lines

Drawing a Cone

Draw a cone by selecting the center point and radius of the cone's base circle, then specifying the radius of the cone's top circle and the cone's height. Cones are drawn as true surfaces in the modeler.

1. Click **Draw>Cone**  .
2. Select the center point of the cone's base circle in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Specify the radius of the cone's base circle by selecting a point on the base circle's circumference. Select the point in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the center point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.
4. Specify the radius of the cone's top circle by selecting a point on its circumference. Select the point by clicking it or typing its coordinates in the **dX**, **dY**, and **dZ** boxes.

   To create an apex, select the same center point as the cone's base circle.

5. Specify the height of the cone by selecting a point on the axis perpendicular to the base circle's plane. Select the point by clicking the point or typing the coordinates in the **dX**, **dY**, and **dZ** boxes.

   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's properties.

6. Click **OK**.

Note	The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see: Technical Notes, "Surface Approximations" and related sections, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Related Topics

Setting the Reference Point

Drawing a Torus

Draw a torus by selecting its center point, major radius, and minor radius. The modeler then sweeps a circle around a circular path. Toruses are drawn as true surfaces in the modeler.

1. Click **Draw>Torus**  .
2. Select the center point of the torus in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Specify the major radius by selecting a point in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the center point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   The major radius determines the diameter of the torus.
4. Specify the minor radius by selecting a point relative to the major radius point.

   The minor radius determines the diameter of the "donut hole".

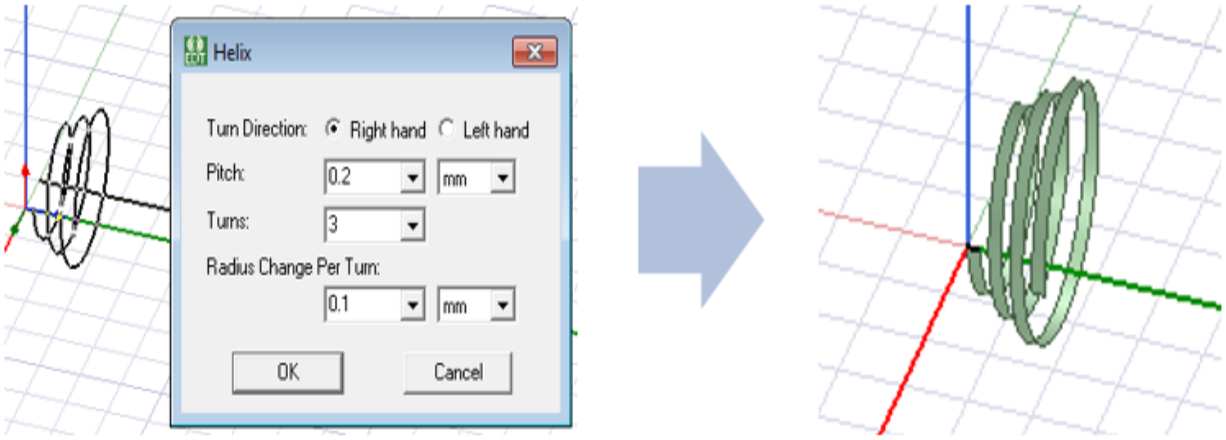
   If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's properties.
5. Click **OK**.

Note	The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see:

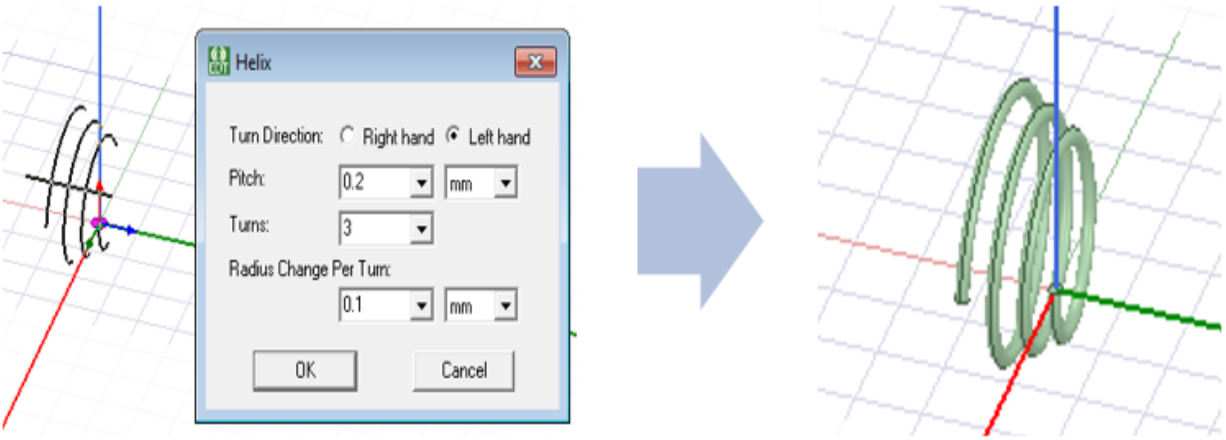
	Technical Notes, "Surface Approximations" and related sections, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Drawing a Helix

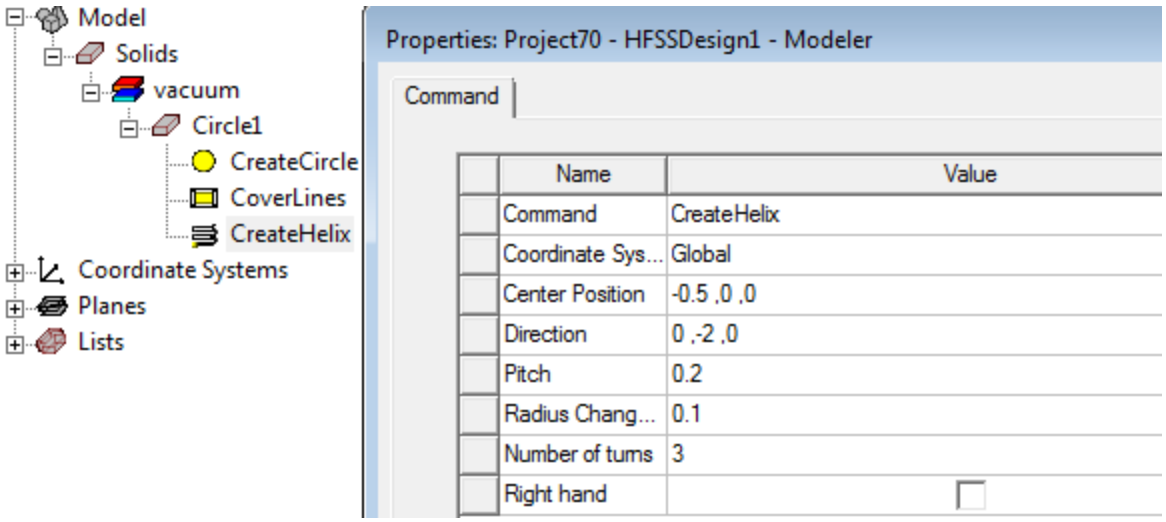
A helix is a 3D spiral object created by sweeping a line or 2D sheet object that you create along a vector that you define. The center of the helix is the defined by the placement of the vector relative to the swept object. You must define the vector so that the swept object will not self-intersect, the rule being, do not draw the vector so that it centers on or overlaps a selected 2D object. After you create the vector, the **Helix** dialog box opens, where you then specify the turn direction, pitch, number of turns, and radius of the turn to define the helix.



Sweeping a 2D object results in a hollow 3D object. Sweeping a 2D sheet object results in a 3D solid object.

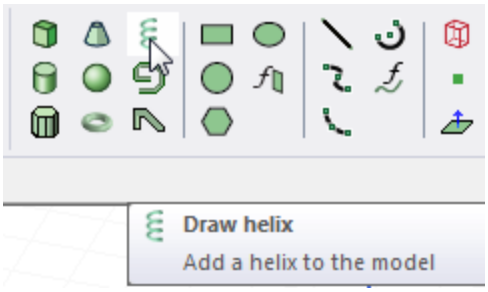


Once you have created a helix, you can changing it by edit the properties of the original object, and by editing the helix command properties for coordinate system, center position (which corresponds to the vector placement), direction, pitch, turns, radius change, and direction.



To create a helix:

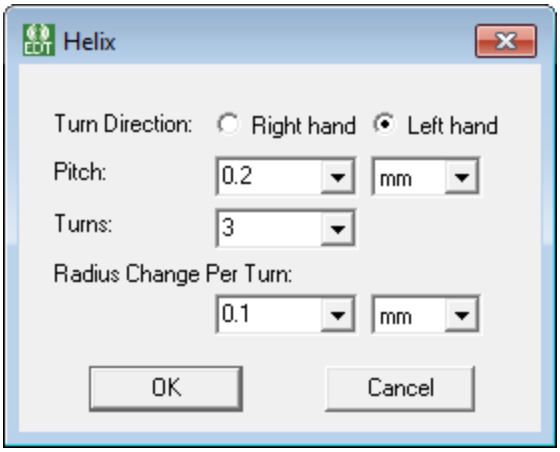
1. Create and select the line or 2D object you want to sweep to form a helix. Selecting a valid object enables the **Helix** icon on the **Draw** tab of the ribbon.
2. Click **Draw**>**Helix** or in the **Draw** tab of the ribbon, click the helix icon:



3. Draw the vector you want to sweep the object around. The two points that describe the vector affect axis location and direction only and not the helix length. (The vector definition corresponds to the **Center Position** and **Direction** properties in the **CreateHelix** command in the **History** tree.) The helix length is determined when you enter the pitch and number of turns in the **Pitch** and **Turns** text boxes. The initial radius of the helix is determined by the axis position relative to the object being swept. It is important that the swept object cannot intersect itself. For a 2D object, you should ensure that the vector you draw does not intersect the interior. If it does, the object cannot be drawn due to self-intersection and you receive a message that "Body could not be created because of invalid parameters."
   a. Select the start point by clicking the point or typing its coordinates in the **X**, **Y**, and **Z** text boxes. (Remember that you can also edit the parameters of a completed helix).

b. Select the endpoint by clicking the point or typing its coordinates relative to the start point in the **dX**, **dY**, and **dZ** boxes.

The **Helix** dialog box appears.



4. For **Turn Direction**, select **Right hand** if the turn direction is clockwise and **Left hand** if the turn direction is counter-clockwise.
5. In the **Pitch** text box, type the distance between each turn in the helix, and click a unit in the pull-down list.
6. In the **Turns** text box, type the number of complete revolutions the object will make along the vector.
7. In the **Radius Change per Turn** text box, type a number for the increase in the radius and select the units from the pull-down list.
8. After you set these values, the selected object is swept along the vector to form a helix. The original object you swept is deleted.
9. Click **OK** to create the Helix. If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, enabling you to modify the object's properties. You can also select the commands for the object and the helix command in the History tree to access and edit their properties.

Note	The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see: Technical Notes, "Surface Approximations" and related sections, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Related Topics

Setting the Reference Point

Drawing a Segmented Helix with Polygon Cross-Section using a User-Defined Primitive

Drawing a Segmented Helix with Rectangular Cross-Section using a User Defined Primitive.

Drawing a Segmented Helix with Polygon Cross-Section Using a User Defined Primitive

Ansys provides you with a DLL to define the parameters of a segmented helix with a polygon cross-section.

1. Click **Draw>User Defined Primitive>SysLib>SegmentedHelix>PolygonHelix**.

   The **Create User Defined Part** dialog box appears. The **Parameters** tab permits you to edit the parameters. An **Info** tab contains information about the user defined primitive, its purpose, the company/author who created it, the date created and the version number.
2. Specify the values for the following parameters:

PolygonSegments	Number of segments in the polygon cross-section. Enter zero (0) for true circle
PolygonRadius	Radius of the polygon cross-section.
StartHelixRadius	The radius of a segmented helix is defined from the helix center of rotation to the center of the helix cross-section at segment transitions. The first and last segments of the helix are half segments. See this figure.
RadiusChange	The radius change per turn of the helix.
Pitch	Distance between helix turns.
Turns	The number of turns in the helix.
SegmentsPerTurn	The number of segments constructing each turn. Enter zero (0) for true curve.
RightHanded	Helix winding direction. Enter non-zero value for right-handed helix.

3. Click **OK**.

Related Topics

Setting the Reference Point

Creating a User Defined Primitive

Drawing a Segmented Helix with Rectangular Cross Section Using a User Defined Primitive

Drawing a Segmented Helix with Rectangular Cross-Section Using a User Defined Primitive

Ansys provides you with a DLL to define the parameters of a segmented helix with a rectangular cross-section.

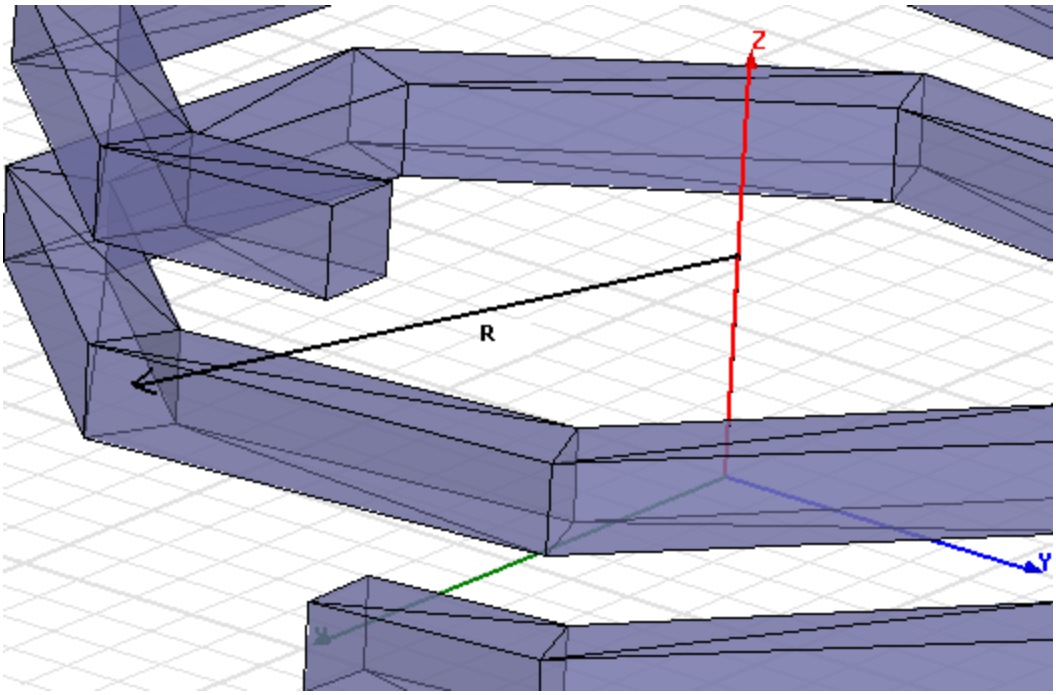
1. Click **Draw>User Defined Primitive>SysLib>SegmentedHelix>RectHelix**.

   The **Create User Defined Part** dialog box appears. The **Parameters** tab permits you to edit the parameters. An **Info** tab contains information about the user defined primitive, its purpose, the company/author who created it, the date created and the version number.

2. Specify the values for the following parameters:

RectHeight	Height of rectangular cross-section.
RectWidth	Width of rectangular cross-section.
StartHelixRadius	The radius of a segmented helix is defined from the helix center of rotation to the center of the helix cross-section at segment transitions. The first and last segments of the helix are half segments. See this figure.
RadiusChange	The radius change per turn of the helix.
Pitch	Distance between helix turns.
Turns	The number of turns in the helix.
SegmentsPerTurn	The number of segments constructing each turn. Enter zero (0) for true curve.
RightHanded	Helix winding direction. Enter non-zero value for right-handed helix.

3. Click **OK**.



Related Topics

Setting the Reference Point

Creating a User Defined Primitive

Drawing a Segmented Helix with Polygon Cross-Section using a User-Defined Primitive

Drawing a Spiral

A spiral is a 2D or 3D spiral object created by sweeping an object around a vector. Sweeping a 1D object results in a 2D sheet object. Sweeping a 2D sheet object results in a 3D solid object.

1. Select the 1D or 2D object you want to sweep to form a spiral.

1. Click **Draw**>**Spiral**  .
2. Draw the vector you want to sweep the object around:
   a. Select the start point by clicking the point or typing its coordinates in the **X**, **Y**, and **Z** text boxes.
   b. Select the endpoint by clicking the point or typing its coordinates relative to the start point in the **dX**, **dY**, and **dZ** boxes.
   The **Spiral** dialog box appears.
3. Select **Right hand** if the turn direction is clockwise and **Left hand** if the turn direction is counter-clockwise.

4. In the **Radius Change** text box, type the difference in radius between each turn of the spiral.

   The radius of the first turn is measured from the center point of the 1D or 2D object you are sweeping to the vector you drew.
5. Click a unit for the radius in the pull-down list.
6. In the **Turns** text box, type the number of complete revolutions the object will make around the vector.

   The selected object is swept around the vector to form a spiral. The original object you swept is deleted.If the Modeler option for editing properties of new primitives is checked, the **Properties** dialog box appears, in which you can modify the object's properties.
7. Click **OK**.

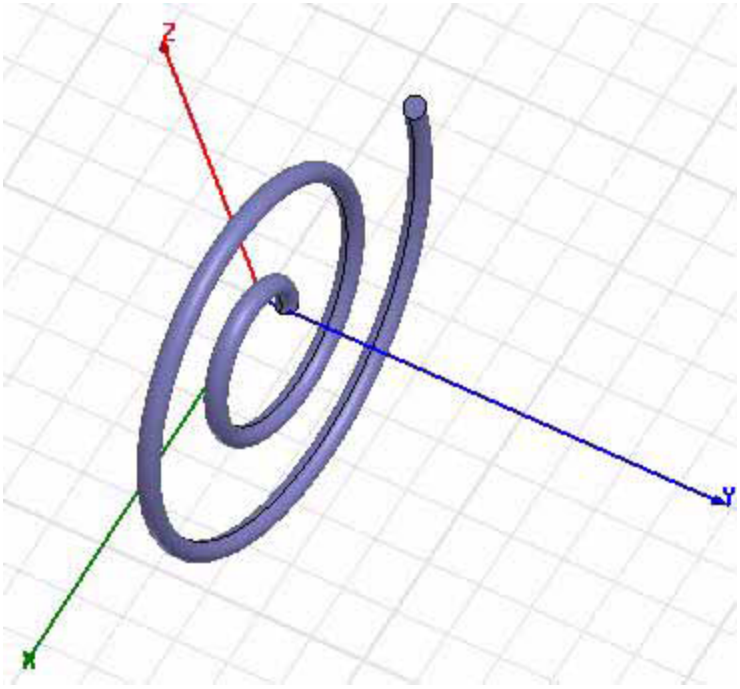


Figure 8-1 This 3D spiral was created from a 2D circle drawn at z = 0. The turn direction was right hand, the radius change was set at 2, and the number of turns was set at 2.

Note	The 3D Geometry Modeler permits drawing of true-curved objects. However, the solution will be obtained with a tetrahedral mesh which conforms to the true surface only within the limits identified by certain mesh settings. The modeler has default settings for this conformance which is a reasonable trade-off between solution speed and solution quality for most objects, but may not be ideal for all such objects. High-aspect ratio curves structures, such as helices with narrow and curved cross-sections, may benefit from user control of the faceting values. For details about these commands see:

	Technical Notes, "Surface Approximations" and related sections, "Modifying Surface Approximations," and "Guidelines for Modifying Surface Approximations"

Related Topics

Setting the Reference Point

Drawing a Spiral Using User Defined Primitives

Drawing a Spiral Using User Defined Primitives

Ansys provides you with a DLL to define the parameters of a rectangular spiral.

1. Click **Draw>User Defined Primitive>SysLib>Examples>RectangularSpiral**.

   The **Create User Defined Part** dialog box appears. The **Parameters** tab permits you to see edit the parameters. An **Info** tab contains information about the user defined primitive, its purpose, the company/author who created it, the date created and the version number.

2. Specify the values for the following parameters:

Xpos	Type the location of the starting point in the X direction.
Ypos	Type the location of the starting point in the Y direction.
TurnSep	Type the separation distance between turns.
Turns	Type the number of complete revolutions the object will make around the vector
Width	Type a value for the width of the spiral.
Height	Type a value for the height of the spiral. If you specify the height as zero, the modeler draws a sheet object.

3. Click **OK**.

   This creates the primitive and displays the **Properties** dialog for the new object.

Hint	To see newly created DLLs, click **Draw>User Defined Primitive>Update Menu.** To see the primitives that you have created, click **Draw>User Defined Primitive>UserLib.**

Related Topics

Setting the Reference Point

Creating a User Defined Primitive

Drawing a Spiral

Drawing a Geodesic Sphere Using a User Defined Primitive

Ansys provides you with a .py script to define the parameters of a geodesic polyhedron and draw a geodesic sphere. A geodesic polyhedron has straight edges and flat faces that approximate a sphere.

1. Click **Draw>User Defined Primitive>Examples>geodesic_sphere**.

   The **User Defined Primitive Operation** dialog box appears. The **Parameters** tab permits you to edit the parameters. An **Info** tab contains information about the user defined primitive, its purpose, the company/author who created it, the date created and the version number.
2. Specify the values for the following parameters:

radius	The radius of the sphere. Enter a positive number and desired unit of measure.
frequency	Indicates how many times each side of the base triangle is subdivided"

3. Click **OK**.

   This creates the primitive and displays the **Properties** dialog for the new object.

Sweeping Objects Around an Axis, Along a Vector, or Along a Path

Use the following three commands (found on the **Draw** menu in the desktop) to sweep a spiral or other object:

- **Draw>Sweep>Around Axis**
- **Draw>Sweep>Along Vector**
- **Draw>Sweep>Along Path**

Drawing a Bondwire

A bondwire is a thin metal wire that connects a metal signal trace with a chip. Please see the topic *Bondwires* in the *Technical Notes* before drawing a bondwire.

1. Click **Draw**>**Bondwire**  .
2. Select the bond pad point in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.
3. Select the lead point by clicking the point or typing the coordinates in the text boxes in the status bar.

   The **Bondwires** dialog box appears.
4. In the **Type** list, click the JEDEC modeling standard shape you want the bondwire to have: **JEDEC 4-point**, **JEDEC 5-point, or Low**.

The **Type** selection changes the dialog bondwire graphic, and shows options for that type.

5. Enter the number of facets in the bondwire in the **No. of Facets** text box.

   The minimum value is 3. The value describes the number of faces that make up the circumference of the bondwire.

6. In the diameter field, specify a diameter value and select the units from the pull-down menu.

7. Enter the height between the bond pad and the top of the loop in the **h1** text box. Include the height's unit of length.
8. The value in the **h2** text box is the height between the bond pad and the lead point. It was calculated by Maxwell based on the lead point you selected. If you modify the value of *h2*, the lead point will be modified.

   Optionally, type a new value in the **h2** text box. Include the height's unit of length.

9. If you selected **JEDEC 5-point or Low** do the following:
   a. Type the angle between the horizontal plane and the wire at the bond pad point in the **alpha** text box.
   b. Type the angle between the horizontal plane and the wire at the lead point in the **beta** text box.
10. Click **OK**.

Related Topics

Setting the Reference Point

Technical Notes:Bondwires

Drawing a Point

Drawing a point object within the problem region enables you to plot fields or perform field computations at that point. Points are always considered non-model objects by the modeler.

1. Click **Draw>Point**  .
2. Select the point in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.

   The point is listed under **Points** in the history tree.

Related Topics

Setting the Reference Point

Modifying Markers on Point Plots

Drawing Non-Model Objects

Drawing a Plane

A plane object is a cutplane through the problem region. You can plot fields or perform field computations on its surface. Planes are always considered non-model objects by the modeler.

1. Click **Draw>Plane**  .
2. Select the origin in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.

   To delete the selected point and start over, press **ESC**.
3. Select a normal point in one of the following ways:
   - Click the point.
   - Type the coordinates of the point relative to the origin in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   The plane is created. Its center point is located at the origin you specified and oriented perpendicular to the normal point you specified. The plane is listed under **Planes** in the history tree.

Note	You only need to draw a plane that does not lie on a predefined xy, yz, and xz plane. Default planes are created on the xy, yz, and xz planes of the global coordinate system as well as any new coordinate system you create.

Related Topics

Drawing Non-Model Objects

Inserting a Line Segment

To create an object with one or more straight line segments after another object, use the **Draw>Line Segment** command.

1. Click **Draw**>**Line Segment>Insert Segment Before** or **Draw**>**Line Segment>Insert Segment After**.
   A submenu appears.
2. Select one of the following:
   - **Straight**
   - **Spline**
   - **3 Point Arc**
   - **Center Point Arc**
3. Select the first point of the line in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the text boxes in the status bar.

4. Select the endpoint of the line by clicking the point or typing the coordinates in the text boxes in the status bar.

   The endpoint serves as the start point for a subsequent line segment.

   To delete all points and start over, press **Esc** or click **Escape Draw Mode** on the shortcut menu.
5. Complete the line in one of the following ways:
   - Double-click the endpoint.
   - Click **Done** on the shortcut menu.
   - Press **Enter**.

   The **Properties** dialog box appears, enabling you to modify the object's attributes.
6. Click **OK**.

Note	While drawing a polyline, you can switch between straight line, arc line, or spline segments using the **Set Edge Type** commands on the shortcut menu.

After drawing a polyline, you can select any segment in the history tree and edit it by changing coordinates, inserting a segment before or after, deleting the start or end point, and making other modifications.

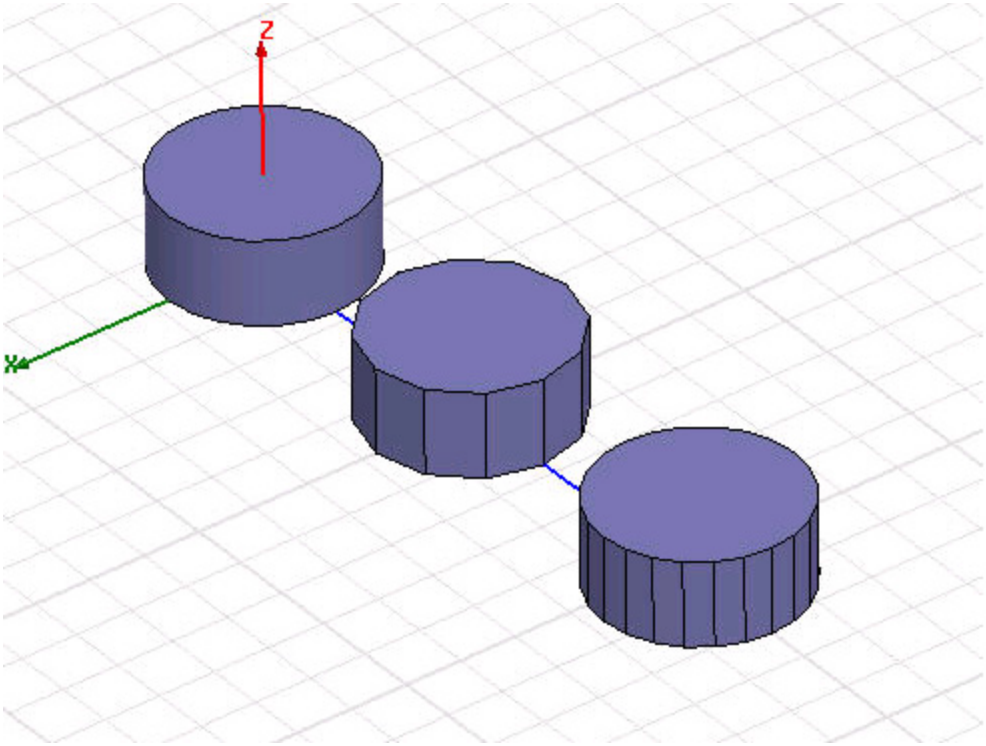
Related Topics

Deleting Polyline Segments

Converting Polyline Segments

Creating Segmented Geometry

For some structures, you may want to create segmented as opposed to smooth (or True) surfaces. The following figure shows a comparison of a cylinder created with true surfaces and with segmented surfaces.



The following model objects can be created as segmented structures:

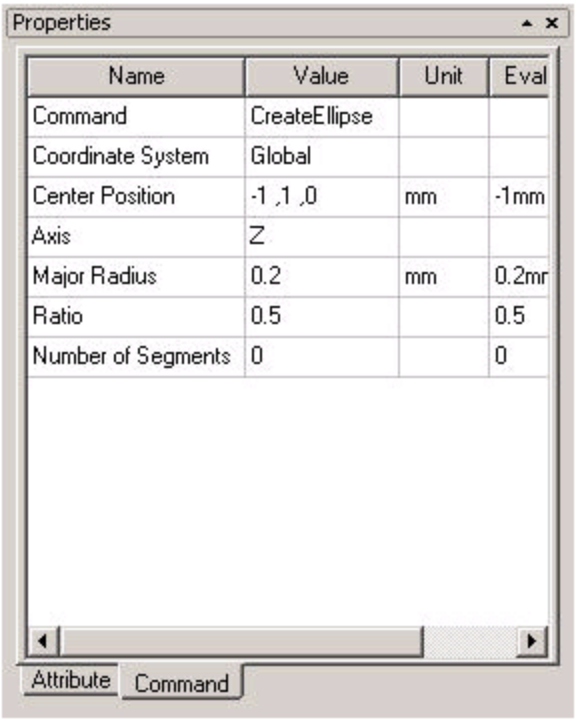
Circle, Ellipse, Cylinder	See Segmented Objects
Polyline, Arc, Line Segment	See Converting Polyline Segments

Segmented Objects

To create segmented circles, ellipses, and cylinders use the **Number of Segments** parameter on the Command Tab of the **Properties** dialog as shown below. To convert an object from true surface to segmented, do the following:

1. Select the **circle**, **ellipse**, or **cylinder** in the modeler window or in the history tree.
2. In the command tab of the properties window (shown docked below), change the **Number**

of Segments to an integer value of three or greater and press **Enter**.



Values of 1 and 2 are not valid values for the circle, ellipse, or cylinder command and will cause an error.

Related Topics

Surface Approximation

Creating Segmented Geometry

Drawing Non-Model Objects

If you want to create an object that does not affect the geometric model, define the object as *non model*. This ensures that the object is used for analysis only; it will not affect the solution process. After drawing the non-model object, assuming it lies in the problem region, you can use it in the reporter as a place on which to plot field quantities. For example, draw a non-model line across the design, then (in the reporter) plot Mag_E on every point along that line.

Following are examples of using non-model objects to analyze a solution:

- Draw a polyline along which to plot fields or perform field computations. Note that when you create a value versus distance plot, by default, the line will be divided into 100 equally spaced points. You can modify the number of points into which the line is divided in the **Edit Sweeps** dialog box.For more information, see *Specifying Variable Values for a Sweep Definition*.
- Draw a rectangle upon which to plot fields or perform field computations.

- Draw a volume box to analyze fields in areas of the problem region that are not occupied by an object or that consist of parts of several objects.
- Draw a plane, which is always a non-model object.
- Draw a point object, which is always a non-model object, in order to plot fields or perform field computations at that point.

What do you want to do?

Switch to non-model drawing mode. Objects you draw in non-model mode will not be included in the solution process.

Modify an existing model object to be a non-model object.

Selecting Non-Model Drawing Mode

To switch to non-model drawing mode:

1. Click **Modeler**>**New Object Type**>**Non Model**.
2. Draw the object.

Related Topics

Changing an Object to Non Model

Drawing Non-Model Objects

Changing an Object to Non-Model

To modify an existing object to be a non-model object:

1. Select the object you want to modify.
2. In the **Properties** dialog box, clear the **Model** option.

   The object will not be included in the solution process. If the object lies in the problem region, you can plot solution quantities on it.

Related Topics

Selecting Non-Model Drawing Mode

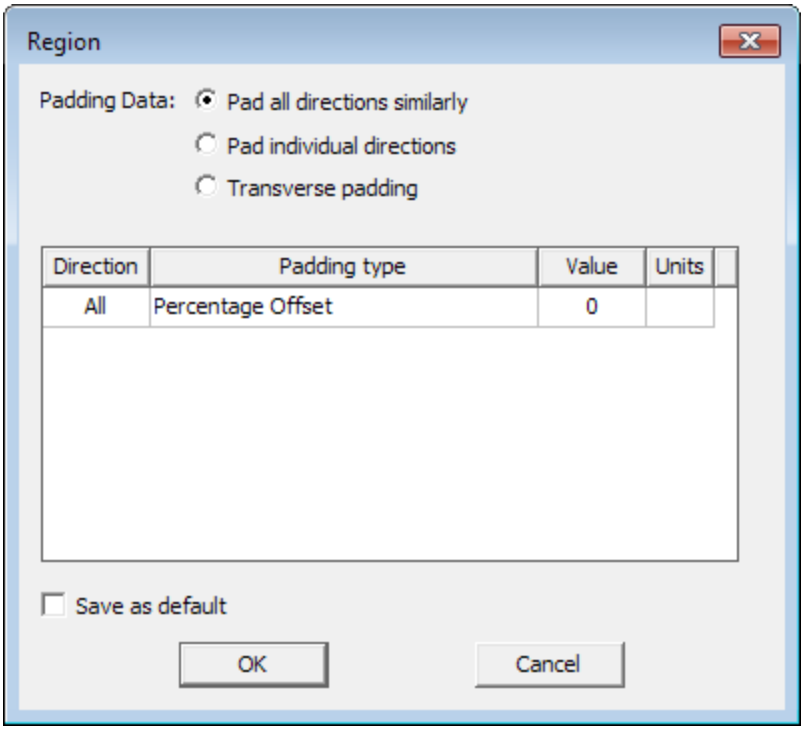
Drawing Non-Model Objects

Drawing a Region

To draw a region encompassing the objects in the current project:

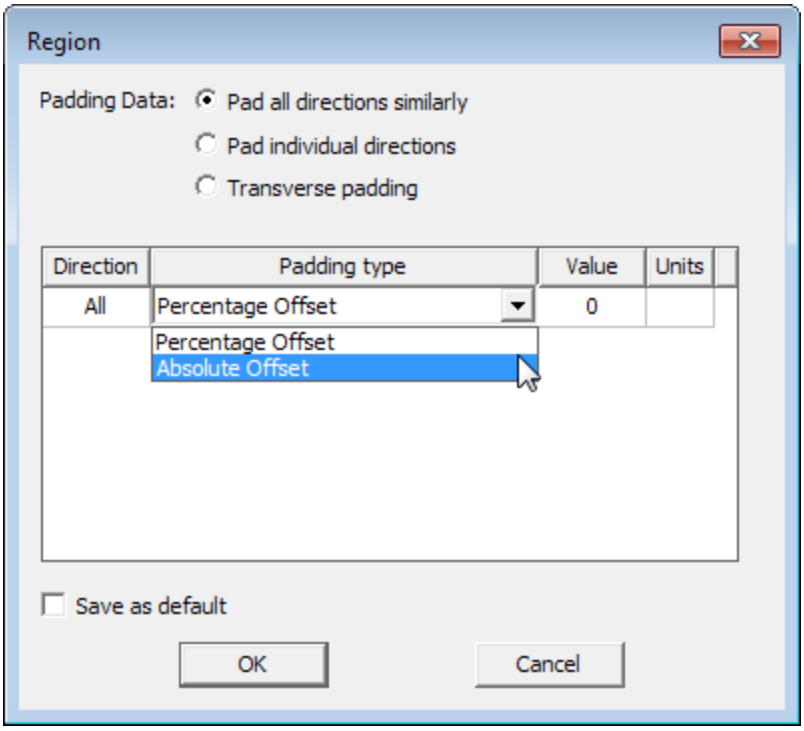
1. Click **Maxwell>Draw>Region** or click the  icon on the tool bar.

   This displays the **Region** dialog box. You can define the region padding as a percentage, relative position, or absolute position.

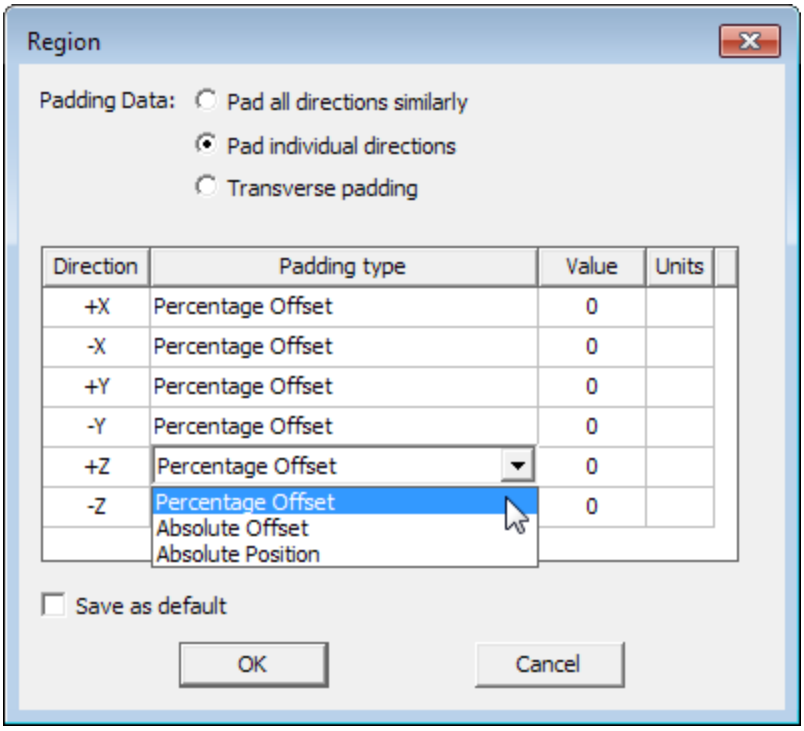


2. For the Padding data, you can select either **Pad all directions similarly**, **Pad individual directions**, or **Transverse padding**.

   If you select **Pad all directions similarly** you can specify the Padding type by selecting either **Percentage Offset** or **Absolute Offset** from the drop down menu, requiring a single Value that affects all directions. If you select **Absolute Offset**, you also specify the units by selecting from a drop down menu.



Selecting **Pad individual directions** displays the Padding Percentage as a table of Positive and Negative X, Y, and Z coordinates, permitting you to specify padding for each direction. In this case, you can specify the Padding type by selecting **Percentage Offset**, **Absolute Offset**, or **Absolute Position** from the drop down menu.



Selecting **Transverse padding** means that padding in one direction is controlled by dimensions in the other two directions. One direction (for example, X) is padded with a user-specified percentage of diagonal length of the other two directions (Y and Z).

3. Specify the Padding values in the fields and select the units from the drop-down list.
4. If desired, click the check box to save the values as Default.
5. Click **OK** to close the dialog and create the region.

   The region is drawn, selected, and displayed in the History tree. It is created using the current coordinate system. The Properties dialog for the region has a Commands tab that shows the coordinate system and Padding values, and the Attributes tab includes properties for Name, Material (Default, vacuum), Solve inside, Orientation, Model, Color, Display Wireframe, and Transparency. You can edit all of these values.

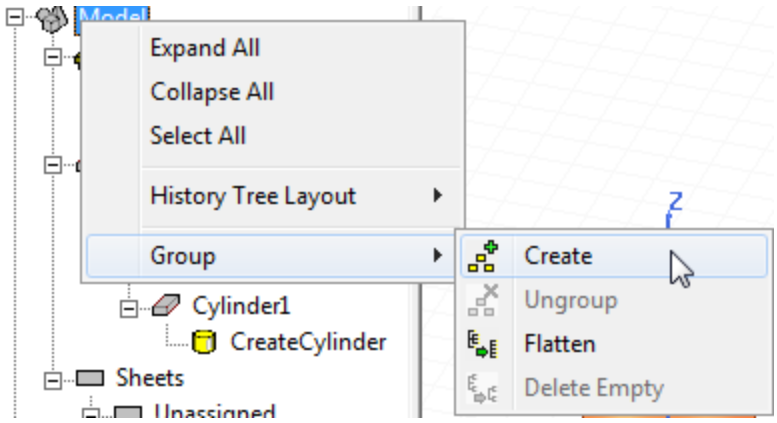
If you try to create a region that does not contain all of the objects in your model, the region is automatically expanded to cover all objects. The region also updates automatically as your geometry changes.

Only one region can be created for a single project using the **Draw>Region** command. If you try to create a second region, the **Properties** window appears for the existing region, allowing you to change operation parameters and attributes.

Group Commands for Modeler Objects

The 3D Modeler allows you to group objects in the History tree. Besides pre-defined folders under **History Tree Layout** for solids, sheets, and material assignments, a set of **Group** commands let

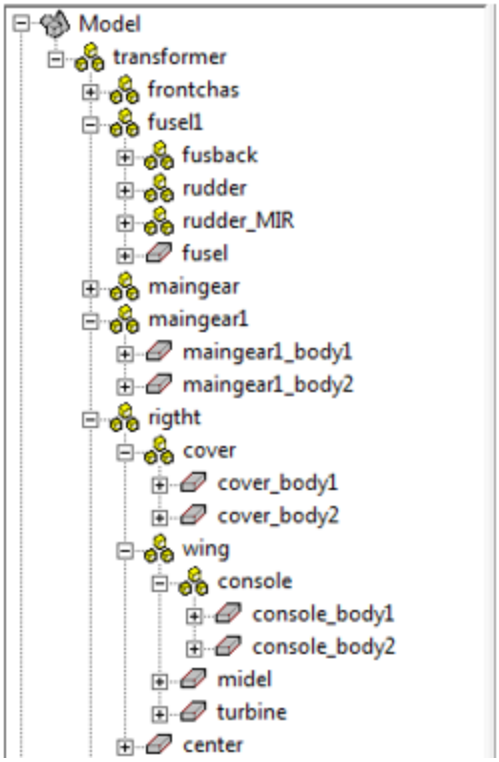
you create groups, un-group objects, flatten a selected group's hierarchy, and delete empty groups.



The **Group** commands support the following features:

- Group objects in the history tree. When objects are grouped, they show up under a sub-item in the history tree.
- Groups can contain sub-groups and sub-group containing further sub-groups.
- Groups permit moving objects from one group to any other group at any time.
- Groups are purely for organization of the history tree. They do not affect the solution in any way.

- Groups permit you to bring in MCAD assemblies and sub-assemblies as groups. See Importing 3D Model Files.



- Groups can contain objects, submodels (for example, UDM, 3D Component, CAD links) and groups. Coordinate systems, planes etc will not have any parent group.

Object groups have the following important differences from 3D components.

- Groups do not encapsulate history of objects. They do not encapsulate parameters used by those objects.
- Groups do not have history tree operations. For example, Arrange operations apply to all objects of the group rather than the group.
- Groups are not independent. Deletion of an object in one group could cause the object in another group to be deleted also.

Accessing the Group Commands

You can access the Group commands in several ways: via the **Modeler>Group** menu, by right-clicking on **Model** in the History tree to see the short-cut menu, the right-click context menu in the Modeler window, and using the Group icons on the Model ribbon.

Group Command Descriptions

Group>Create works with selection or no selections. If there are no selections, an empty group is created under model. If there are selections, then a new group is created under same group as all the selections, provided all selections are under one group. Once a group is created, all selections move under new group. If selections are under different groups, then a new group is created

under model. Selections could be objects, groups, 3D Components, or User Defined Model. The **Group> Create** command tries to find a more suitable group name where possible based on names of all selections. If not, default group names are Group1, Group2 and so on. You can edit group names later.

Group>Ungroup is enabled only when you select one or more groups. Upon ungroup, all the contents of the selected group move under group's parent and the selected group is deleted. Note that **Group>Ungroup** is not recursive; it only affects the selected group is and all its children groups remain intact. Ungroup is also different from **Delete** (under **Edit>Delete**) which deletes a selected group and everything under it.

Group>Flatten Flatten Group is enabled when you select a group selection. You can select multiple groups at different levels. All of the selected group's contents (that is, objects, sub-groups, and so forth) come directly under selected group. In a sense this is a recursive ungroup operation as all the children and grandchildren groups of selected group are ungrouped.

Group>Delete Empty deletes all empty groups under the selected group. If a selected group is empty it is deleted. This command is helpful to clean up empty groups after you have moved objects from one group to another.

Operations on Groups

Edit/Delete: Delete operations delete group and all the children, including sub-groups, under the group.

Copy and Paste of Objects in Groups

When you copy and paste objects with groups to another design, group information is carried over, that is, new group in target design is created. If a group with the exact name exists, then the objects are moved under that group and no new group is created. When pasted in the same design, the pasted object moves under same parent group as original object.

Copy and Paste of Groups

When you copy and paste one or more groups in same design or to another design, a new group is created with unique name derived from original group name. Everything under that group, including any sub-groups, is copy/pasted as well.

Copy and Paste of Groups and Objects

If you multi-select a few objects and groups, all of them are pasted correctly. If you select a group and a few objects under that group, copy and paste includes the complete group structure. This means that selection of objects under the selected group is ignored. If you want to only copy and paste few objects from group, you should only select those objects and NOT select the group.

Arrange Operations on Groups

You can select group to enable arrange operations like move, rotate and mirror. Note that arrange operations works directly on objects under selected group. So after arrange operation is done, the History tree shows the arrange operation under every object of that group.

Duplicate Operations on Groups

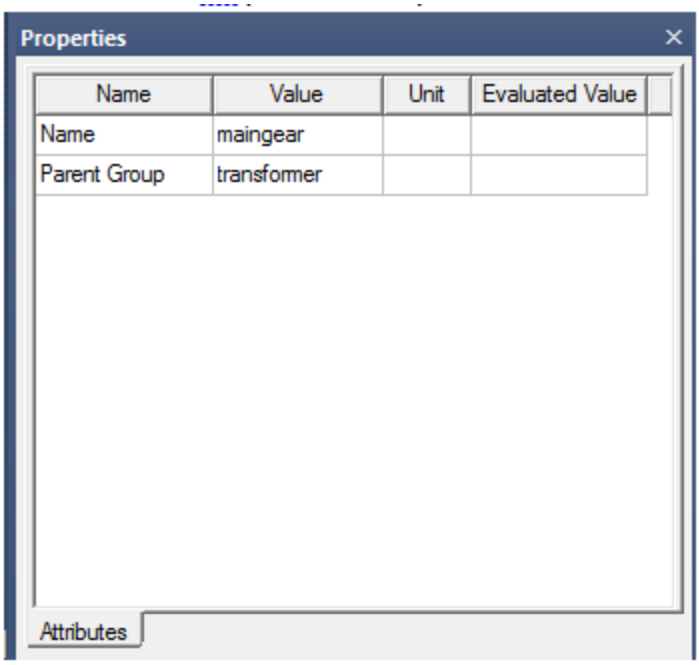
You can select groups to enable duplicate operations. Afterward, the History tree shows a duplicate operation under every object of that group. Newly created objects appear under a new group. There will be new group for every instance of duplicate.

Other Operations on Groups and Objects in Groups

Modeling operations that create new objects, such as Create from Face or Edge, as well as Simplify and Separate. The newly created objects appear under a new group created under the original object's group, based on the tool option settings. See 3D Modeler Options: Group Options.

Group Properties

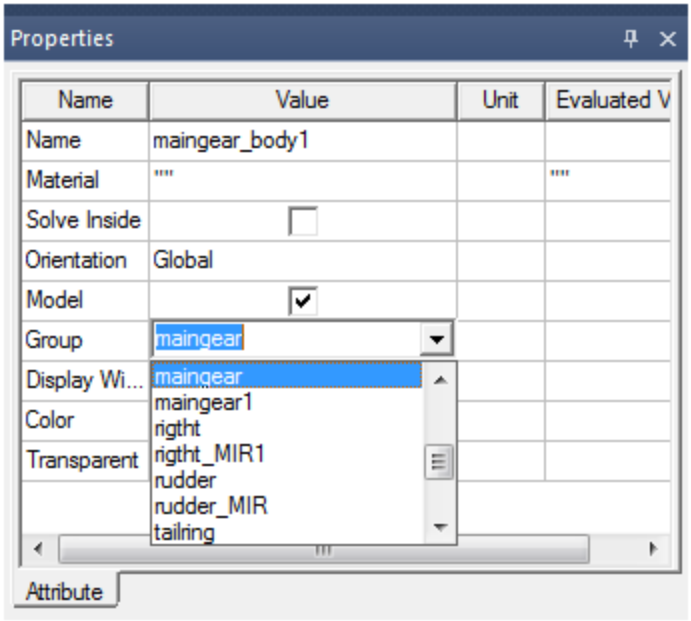
A Group's Property tab is shown when you select a group in the history tree. The Group's properties include Name and parent Group.



Group Property Window

Object, group and submodel properties have a Group property. You can edit the Group Property and this provides another way of setting an object's (or group's) group. You can select any of the

existing groups listed or create a new group.



Group assignment could be changed in Object's property window

Related Topics

Select Objects in the History Tree

User Customization through User Defined Primitives (UDPs)

User Defined Primitives (UDPs) allow users to add customized geometric modeling commands to the Maxwell Desktop. UDPs are C++ compiled or Python libraries that can be added to the desktop interface and shared between users with common modeling needs.

To create a UDP, see Creating a User Defined Primitive for requirements and the procedure for building a proper DLL or Python library.

In order to share UDPs between users, an existing DLL or Python library may be copied into the **userlib>User Defined Primitives** subdirectory which can be found in the Maxwell installation directory. Placing an appropriately constructed DLL or Python library in this subdirectory will automatically add a new menu item in the **Draw>User Defined Primitives** menu to allow access to the UDP.

If desired, users can also create a **User Defined Primitives** subdirectory in their **PersonalLib** directory, and store UDPs there. The **PersonalLib** directory path is specified on the **Directories** panel under **General** in the **Options** dialog.

Refer to *Maxwell User Defined Primitives* for additional information.

Related Topics

Creating a User Defined Primitive

Drawing a Segmented Helix with Polygon Cross-Section Using a User Defined Primitive

Drawing a Segmented Helix with Rectangular Cross-Section Using a User Defined Primitive

Drawing a Spiral Using User Defined Primitives

User Defined Primitives (UDPs) for Motor Geometries

Parameters for RMxprt User Defined Primitives

Creating a User Defined Primitive

The modeler allows you to generate user-defined primitives, primitives customized to suit any application. User-defined primitives are accessed using DLLs that you build and compile. When user defined primitives exist in your UserLib or Personal directory (given the paths specified in **Tools>General Options**Project Options tab), they appear in the **Draw>User Defined Primitives>** menu. Newly created UDPs will appear after a restart, or **Draw>User Defined Primitive>Update Menu.**

Example C++ source and header files that can be used to generate DLLs are located in the **UserDefinedPrimitives/Examples** subdirectory under the Ansys Electromagnetics Suite installation directory.

As an example, create the primitive **myUDP.dll** using Microsoft Visual C++ Developer Studio:

1. Create a directory to store all of the workspace information, call it `UDPDir`.
2. Use the sample workspace **RectangularSpiral.dsw** as a template:
   a. Copy **RectangularSpiral.dsw** and **RectangularSpiral.dsp** from the **`UserDefinedPrimitives/Examples`** directory to this new directory.
   b. Make sure the new files have write permissions.
   c. Rename the files to **myUDP.dsw** and **myUDP.dsp** respectively.
   d. Open the **.dsw** and **.dsp** files in a text editor, and replace every occurrence of **RectangularSpiral** with **myUDP**.
   e. Save **myUDP.dsp** and **myUDP.dsw**.
3. In the **`UDPDir`** directory, create a **`Headers`** subdirectory.
4. Copy the **UserDefinedPrimitiveStructures.h** and **UserDefinedPrimitiveDLLInclude.h** files from the **`UserDefinedPrimitives/Headers`** directory.

Note	The header files include information on the methods that are available for use in your source code. They must be included when you compile the DLL.

5. In the **`UDPDir`** directory, create a **`Sources`** subdirectory.
6. Use the sample source file **RectangularSpiral.cpp** as a template:
   a. Copy **RectangularSpiral.cpp** from the **`UserDefinedPrimitives/Examples`** directory to this new directory.

b. Make sure the new file has write permission.
c. Rename the file to **myUDP.cpp**.

The resulting directory structure will appear similar to the following:

```
UDPDir/
        myUDP.dsw
        myUDP.dsp
        Headers/
          UserDefinedPrimitiveDLLInclude.h
          UserDefinedPrimitiveStructures.h
        Sources/
          myUDP.cpp
```

7. Open **myUDP.dsw** using Microsoft Visual C++ Developer Studio, and edit the source code to create your desired primitive. You may also add additional headers and source files as appropriate.

   The UDP dll contains a data structure called UDPPrimitiveTypeInfo. This contains information about the udp, its purpose, company/author who created it, date created and the version number. When you select a primitive from your library, you see the **Create Primitive** dialog with a **Parameters** tab for setting the parameters, and an **Info** tab with the information from this data structure.
8. Build **myUDP.dll** using the **Win32 Release** configuration.
9. Copy the resulting file **myUDP.dll** to the **`[ProductInstallationPath]/userlib/UserDefinedPrimitives`** directory.

   Optionally, copy the resulting file to the `UserDefinedPrimitives` directory in your PersonalLib. The **PersonalLib** directory path is specified on the **Directories** panel under **General** in the **Options** dialog.
10. To view your primitives, click **Draw>User Defined Primitive>Update Menu** and then click **Draw>User Defined Primitive>UserLib**or **Draw>User Defined Primitive>PersonalLib**.

Note	On Linux, you may use the same example directory structure, source, and header files to build and compile a shared library using C++. The resulting shared library needs to be placed in the same **`/userlib/UserDefinedPrimitives`** directory. As with the Windows DLL, the compiled library will work only on the operating system on which it was built.

Related Topics

User Defined Primitives for Motor Geometries

Drawing a Spiral Using User Defined Primitives

Drawing a Segmented Helix with Polygon Cross-Section Using a User Defined Primitive

Drawing a Segmented Helix with Rectangular Cross-Section Using a User Defined Primitive

Specifying UDP Parameters

The **User Defined Primitive Operation** dialog box allows you to set the specific parameters required for a UDP.

In the **Parameters** tab, a list of the parameters associated with the UDP is displayed along with the value, unit, evaluated value, and description for each parameter. To change a parameter:

1. Select the **Value** cell in the row corresponding to the parameter to be changed. The cell becomes editable. Enter the desired **Value** for the parameter.
2. Click on the cell under **Unit** to obtain a list of the available units for the parameter. Select the desired **Unit**.
3. When the desired changes have been made, click **OK** to create the geometry defined by the UDP and the parameters you have entered.

The **Info** tab provides information about the UDP.

Related Topics

Drawing a Segmented Helix with Polygon Cross-Section Using a User Defined Primitive

Drawing a Segmented Helix with Rectangular Cross-Section Using a User Defined Primitive

Drawing a Spiral Using User Defined Primitives

Parameters for RMxprt User Defined Primitives

Updating the User Defined Primitives Menu

When new User Defined Primitives have been created or provided for your use, they should be placed in the **Maxwell/userlib/UserDefinedPrimitives** directory. Optionally, user defined primitives can be placed in your **PersonalLib**. The **PersonalLib** directory path is specified on the **Directories** panel under **General** in the **Options** dialog.The **Update Menu** command scans the disk and creates menu items for new UDPs. To create menu items for new UDPs:

1. Place new UDPs in either of the directories listed above. Organizing UDPs using subdirectories is permitted.
2. Select **Draw>User Defined Primitive>Update Menu**.

   The new UDPs may now be run directly from the **Maxwell** menu.

Related Topics

User Customization Through User Defined Primitives

Creating a User Defined Primitive

User Defined Primitives (UDPs) for Motor Geometries

When designing a motor or other machine in either Maxwell or RMxprt, multiple stator, rotor, and pole geometries are available.

Related Topics

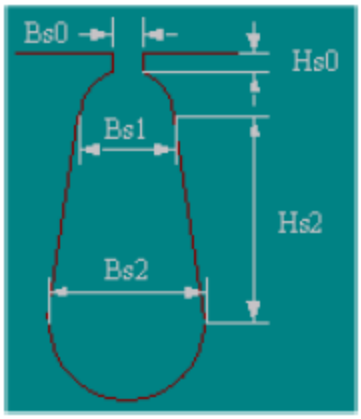
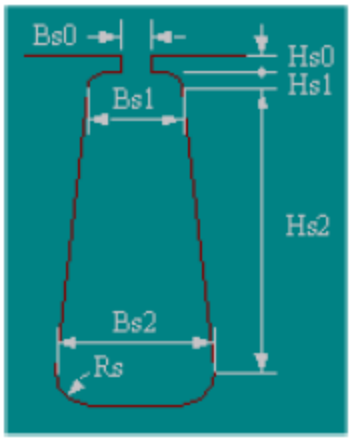
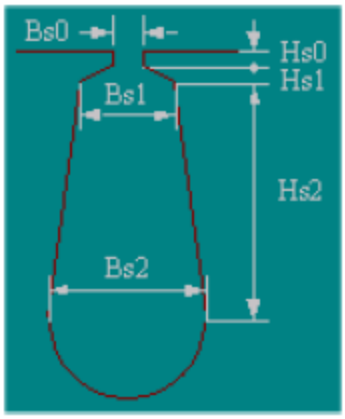
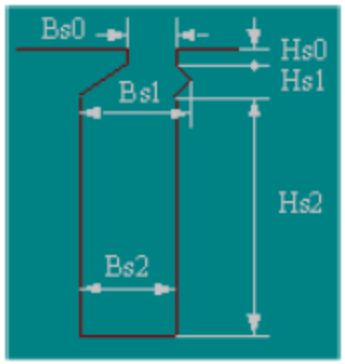
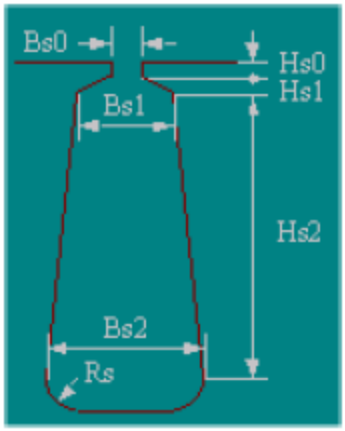
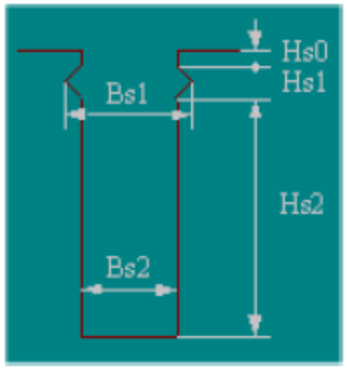
Parameters for RMxprt User Defined Primitives

Stator Geometries

Rotor Geometries

Pole Geometries

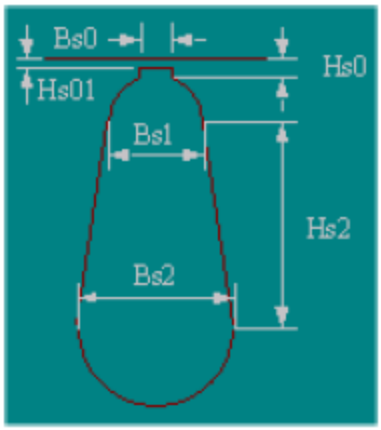
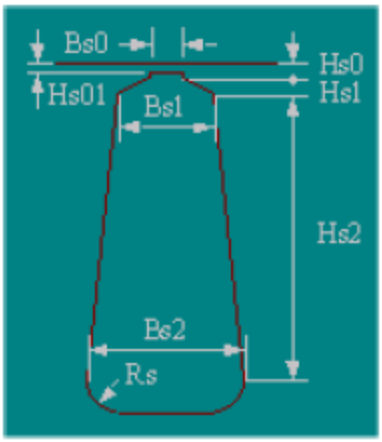
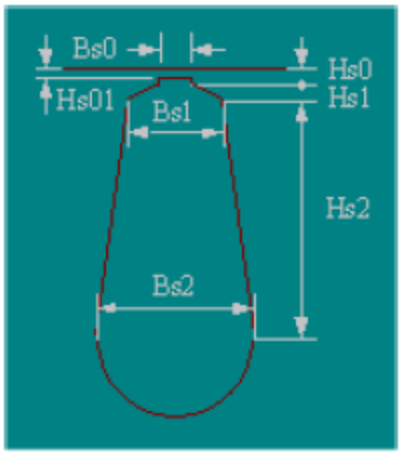
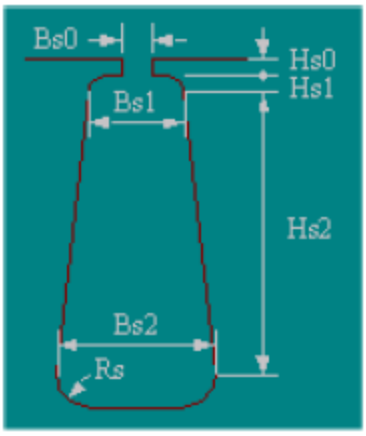
Stator Geometries

Slot Type	Slot Dimensions	Slot Type	Slot Dimensions
1		4	
2		5	
3		6	

Related Topics

Parameters for RMxprt User Defined Primitives
Specifying UDP Parameters

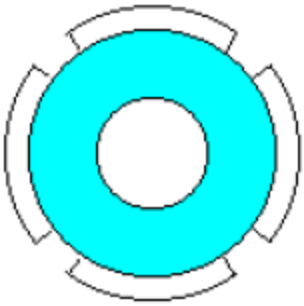
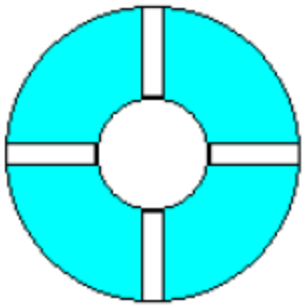
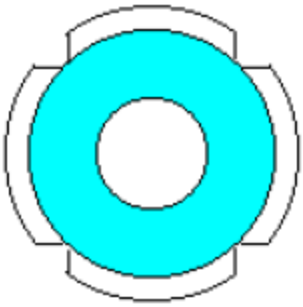
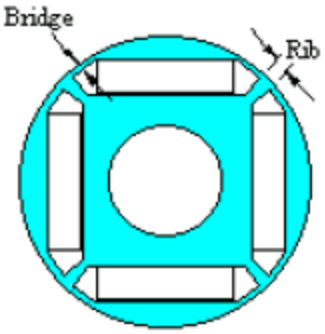
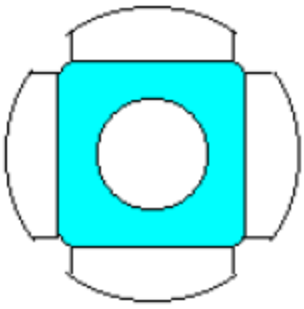
Rotor Geometries

Slot Type	Slot Dimensions	Slot Type	Slot Dimensions
1		3	
2		4	

Related Topics

Parameters for RMxprt User Defined Primitives
Specifying UDP Parameters

Pole Geometries

Slot Type	Slot Dimensions	Slot Type	Slot Dimensions
1		4	
2		5	
3			

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

Parameters for RMxprt User Defined Primitives

In the **Draw/User Defined Primitive/syslib/RMxprt** menu, the following User Defined Primitives exists for creating elements of motors.

Select the primitive of interest to see the parameters required.

Band	**LapCoil**	**SRMCore**
ClawPoleCore	**LinearMCore**	**SynRMCore**
ConCoil	**NonSalientPoleCore**	**TransCoil**
DCMCore	**PMCore**	**TransCore**
DiskCoil	**PMDamperCore**	**UnivMCore**
DiskPMCore	**RacetrackSlotCore**	**VentSlotCore**
DiskSlotCore	**SalientPoleCore**	**WaveCoil**
DoubleCage	**SlotCore**	
IPMCore	**SquirrelCage**	

RMxprt/Band

Property	Description
DiaGap	Band diameter in gap center, DiaGap < DiaYoke for outer band
DiaYoke	Band diameter on yoke side, DiaYoke < DiaGap for inner band
Length	Band Length
SegAngle	Angle per segment of band (0.1 ~ 5 degrees)
Fractions	Number of circumferential fractions, 1 for circular region
InfoCore	0: band; 100: outer region or shaft

Related Topics

Parameters for RMxprt User Defined Primitives

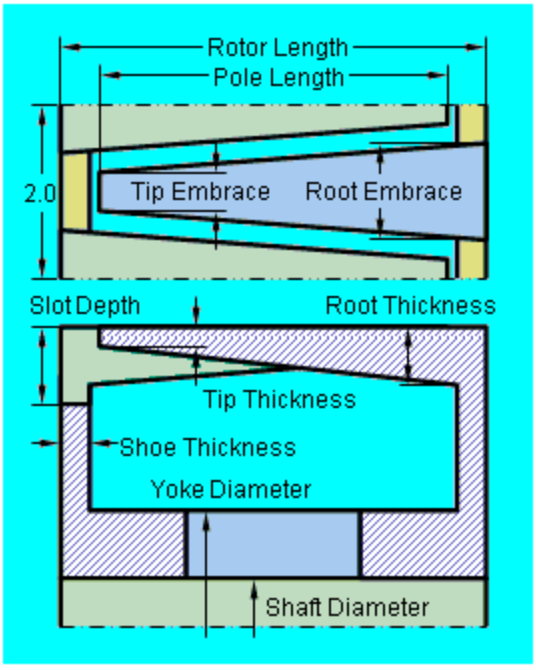
Specifying UDP Parameters

RMxprt/ClawPoleCore

See graphic below for definitions.

Property	Description

DiaGap	Core diameter on gap side, DiaGap < DiaShaft for outer cores
DiaShaft	Core diameter on shaft side, DiaShaft < DiaGap for inner cores
Length	Core Length
Poles	Number of poles
EmbraceTip	Embrace of pole tip
EmbraceRoot	Embrace of pole root
ThickTip	Thickness of pole tip
ThickRoot	Thickness of pole root
ThickShoe	Thickness of side shoes
DepthSlot	Depth of slot between two poles
ThickYoke	Thickness of yoke
LengthPole	Length of pole from tip to tip
LengthMag	Length of magnet or the second air-gap
LenRegion	Region length
InfoCore	0: core; 1: core&coil; 2: coil; 3: magnet; 100: region



Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/ConCoil

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores
Length	Core Length
Skew	Skew angle in core length range
Slots	Number of slots
SlotType	Slot Type: 1 to 7
Hs0	Slot opening height
Hs1	Slot wedge height
Hs2	Slot body height
Bs0	Slot opening width
Bs1	Slot wedge maximum width
Bs2	Slot body bottom width, 0 for parallel teeth
Rs	Slot body bottom fillet
FilletType	0: a quarter circle; 1: tangent connection
LayerLoc	0: whole; 1: middle; 2: top; 3: bottom
CoilPitch	Coil pitch measured in slots
EndExt	One-side end extended length
LayerExt	Span layer extension in the axial direction
LayerDiff	Span layer difference in the radial direction
AltEnd	0: same end layers; 1: alternate end layers
LenRegion	Region length
InfoCoil	0: coil; 1: terminal1; 2: terminal2; 100: region

Related Topics

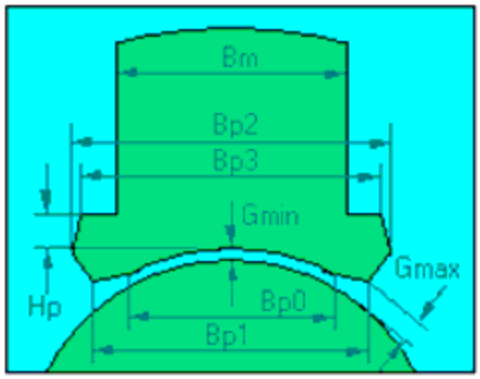
Parameters for RMxprt User Defined Primitives
Specifying UDP Parameters

RMxprt/DCMCore

See graphics below for definitions.

Property	Description

DiaGap	Core diameter on gap side, or Dmin, the minimum inner diameter
DiaYoke	Core diameter on yoke side, or frame outer diameter
Length	Main pole length, 0 for 2D geometry
Skew	Skew angle in pole length range
FrameWidth	Overall width of a racetrack frame
FrameThick	Frame thickness
FrameLength	Frame length
Poles	Number of poles
PoleType	Pole Type: 1 or 2
Dmax	Diameter of shoe tip with maximum air gap length
Bp0	Pole arc width with uniform air gap, 0 for eccentric air gap
Bp1	Total pole width (the width between shoe tips)
Bp2/Rp0	Bp2 (max shoe width for pole type 1), or Rp0 (shoe fillet radius for pole type 2)
Bp3/Rp1	Bp3 (min shoe width for pole type 1), or Rp1 (pole fillet radius for pole type 2)
Hp	Pole body height
Bm	Pole body width
FieldWndgs	Number of field windings, 2 for both series and shunt windings
EndExt	Coil one-side end extended length
SlotsPerPole	Compensating slots per pole distributed under pole arc surface
Bc0	Opening width of compensation slots
Hc0	Opening height of compensating slots
Bc2	Width of compensating slots
Hc2	Height of compensating slots
CmpEndExt	One-side end extended length of compensating coils
ComPoleWidth	Width of the commutating poles
ComPoleHeight	Height of the commutating poles
ComPoleLength	Length of the commutating poles
ComShoeWidth	Shoe width of the commutating poles
ComShoeHeight	Shoe height of the commutating poles
ComShoeLength	Shoe length of the commutating poles
ComGap2	Air gap length between commutating poles and the frame
CmtEndExt	One-side end extended length of commutating coils
LenRegion	Region length
InfoCore	0: core&coil; 1: poles; 2: frame; 3: com poles; 4: shunt coil; 5: series coil; 6: com coil; 100: region
InfoTerm	0: whole coil; 1: terminal1; 2: terminal2



Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/DiskCoil

Property	Description
DiaOuter	Core outer diameter
DiaInner	Core inner diameter
Thickness	Core thickness
Gap	Gap between the core and xy plane, <0 for lower core
Skew	Skew angle
Slots	Number of slots
SlotType	Slot type: 1 to 7
Hs0	Slot opening height
Hs1	Slot wedge height
Hs2	Slot body height
Bs0	Slot opening width
Bs1	Slot wedge maximum width
Bs2	Slot body bottom width. 0 for parallel teeth
Rs	Slot body bottom fillet
FilletType	0: a quarter circle; 1: tangent connection

Layers	Number of winding layers
CoilPitch	Coil pitch measured in slots
EndExt	One-side end extended length
EndClr	Clearance between two end spans; 0 for no span
InfoCoil	0: all coils; 1: one coil only
SegAngle	Angle per segment of end span: 5~15 degrees

Related Topics

Parameters for RMxprt User Defined Primitives
Specifying UDP Parameters

RMxprt/DiskPMCore

Property	Description
DiaOuter	Core outer diameter
DiaInner	Core inner diameter
Thickness	Core thickness
Gap	Gap between the core and xy plane, <0 for lower core
Skew	Skew angle
Poles	Number of poles
Embrace	Pole embrace
ThickMag	Magnet thickness
InfoCore	0: core only; 1: all PMs; 2: one PM only

Related Topics

Parameters for RMxprt User Defined Primitives
Specifying UDP Parameters

RMxprt/DiskSlotCore

Property	Description
DiaOuter	Core outer diameter
DiaInner	Core inner diameter
Thickness	Core thickness

Gap	Gap between the core and xy plane, <0 for lower core
Skew	Skew angle
Slots	Number of slots
SlotType	Slot type: 1 to 6
Hs0	Slot opening height
Hs01	Slot closed bridge height
Hs1	Slot wedge height
Hs2	Slot body height
Bs0	Slot opening width
Bs1	Slot wedge maximum width
Bs2	Slot body bottom width. 0 for parallel teeth
Rs	Slot body bottom fillet
FilletType	0: a quarter circle; 1: tangent connection
HalfSlot	0: symmetric slots; 1: half slots
RingLength	One-side radial ring length
RingHeight	Axial ring height
InfoCore	0: core only; 1: squirrel cage only

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/DoubleCage

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores
Length	Core Length
Skew	Skew angle in core length range
Slots	Number of slots
SlotType	Slot Type: 1 to 4
Hs0	Slot opening height
Hs01	Slot closed bridge height
Hs1	Slot wedge height
Hs2	Slot body height

Bs0	Slot opening width
Bs1	Slot wedge maximum width
Bs2	Slot body bottom width, 0 for parallel teeth
Rs	Slot body bottom fillet
FilletType	0: a quarter circle; 1: tangent connection; 2 & 3: arc bottom
HalfSlot	0: symmetric slots; 1: half slots
BarEndExt	One-side bar end extended length
RingLength	One-side radial ring length
RingHeight	Radial ring height
RingDiaGap	Ring diameter on gap side
DoubleCage	0: normal squirrel cage; 1: double squirrel cage
BSlotType	Bottom slot type: 1 to 4
BHs0	Slot opening height
BHs1	Slot wedge height
BHs2	Slot body height
BBs0	Slot opening width
BBs1	Slot wedge maximum width
BBs2	Slot body bottom width, 0 for parallel teeth
BRs	Slot body bottom fillet
CastRotor	0: insert-bar; 1: cast-rotor
LenRegion	Region length
InfoCoil	0: bar & rings; 1: bars; 2: rings; 100: region

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/IPMCore

Property	Description
DiaGap	Core diameter on gap side, or outer diameter
DiaYoke	Core diameter on yoke side, or inner diameter
Length	Core Length
Poles	Number of poles
PoleType	Pole Type: 1 or 3

D1	Limited diameter of PM ducts
O1	Bottom width for separate or flat-bottom duct
O2	Distance from duct bottom to shaft surface
B1	Duct thickness
Rib	Rib width
HRib	Rib height
DminMag	Minimum distance between side magnets
ThickMag	Magnet thickness
WidthMag	Total width of all magnet per pole
LenRegion	Region length
InfoCoil	0: core; 1: magnets; 2: ducts; 100: region

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/LapCoil

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores
Length	Core Length
Skew	Skew angle in core length range
Slots	Number of slots
SlotType	Slot Type: 1 to 7
Hs0	Slot opening height
Hs1	Slot wedge height
Hs2	Slot body height
Bs0	Slot opening width
Bs1	Slot wedge maximum width
Bs2	Slot body bottom width, 0 for parallel teeth
Rs	Slot body bottom fillet
FilletType	0: a quarter circle; 1: tangent connection
Layers	Number of winding layers
CoilPitch	Coil pitch measured in slots

EndExt	One-side end extended length
SpanExt	Axial length of end span; 0 for no span
SegAngle	Angle per segment of end span: <5 for true-surface end span
LenRegion	Region length
InfoCoil	0: winding; 1: coil; 2: terminal1; 3:terminal2; 100:region

Related Topics

Parameters for RMxprt User Defined Primitives
Specifying UDP Parameters

RMxprt/LinearMCore

Property	Description
WidthCore	Core width in motion direction
ThickCore	Core thickness
Length	Core length
SlotPitch	The distance between two slots
Slots	Number of slots
SlotType	Slot Type: 1 to 7
Hs0	Slot opening height
Hs1	Slot wedge height
Hs2	Slot body height
Bs0	Slot opening width
Bs1	Slot wedge maximum width
Bs2	Slot body bottom width, 0 for parallel teeth
Rs	Slot body bottom fillet
Layers	Number of winding layers
CoilPitch	Coil pitch measured in slots
EndExt	One-side end extended length
SpanExt	Axial length of end span; 0 for no span
InfoCore	0: core only; 1: core & all coils; 2: one coil only

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/NonSalientPoleCore

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores
Length	Core Length
Skew	Skew angle in core length range
IndexingSlots	Number of indexing slots for slot pitch calculation
ActualSlots	Number of actual slots: ActualSlots <= IndexingSlots
SlotType	Slot Type: 1 to 7
Hs0	Slot opening height
Hs1	Slot wedge height
Hs2	Slot body height
Bs0	Slot opening width
Bs1	Slot wedge maximum width
Bs2	Slot body bottom width, 0 for parallel teeth
Rs	Slot body bottom fillet
FilletType	0: a quarter circle; 1: tangent connection; 2 & 3: arc bottom
Poles	Number of poles
TangentDucts	Number of surface tangential vent ducts
TDuctWidth	Width of surface tangential vent ducts
TDuctDepth	Depth of surface tangential vent ducts
TDuctPitch	Pitch of surface tangential vent ducts
AxialDucts	Number of axial vent ducts per pole
ADuctWidth	Width of axial vent ducts in main tooth
ADuctDepth	Depth of axial vent ducts in main tooth
LenRegion	Region length
InfoCore	0: core; 100: region

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/PMCore

Property	Description
DiaGap	Core diameter on gap side, or outer diameter
DiaYoke	Core diameter on yoke side, or inner diameter
Length	Core Length
Poles	Number of poles
PoleType	Pole Type: 1 or 5
Embrace	Pole embrace (not for type 4)
ThickMag	Max thickness of magnets
WidthMag	Magnet width (for types 4 & 5)
Offset	Pole arc offset (for types 1, 2, & 3)
Bridge	Bridge thickness (for type 5 only)
Rib	Rib width (for type 5 only), Rib=0 for rectangular ducts
LenRegion	Region length
InfoCoil	0: core; 1: magnets; 2: magnet; 100: region

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/PMDamperCore

Property	Description
DiaGap	Core diameter on gap side, or outer diameter
DiaYoke	Core diameter on yoke side, or inner diameter
Length	Core Length
Skew	Skew angle in core length range
Slots	Number of damper slots per pole
SlotType	Slot type: 1 to 4
Hs0	Slot opening height
Hs01	Slot closed bridge height
Hs1	Slot wedge height
Hs2	Slot body height
Bs0	Slot opening width

Bs1	Slot wedge maximum width
Bs2	Slot body bottom width, 0 for parallel teeth
Rs	Slot body bottom fillet
FilletType	0: a quarter circle; 1: tangent connection; 2 & 3: arc bottom
Poles	Number of poles
PoleType	Pole Type: 1 or 8
D1	Limited diameter of PM ducts
O1	A locating dimension of PM ducts, for pole types 3, 4, 7 & 8
O2	A locating dimension of PM ducts, for pole types 1, 3, 5, 6, 7 & 8
B1	Barrier width, for pole types 1, 2 & 3
Rib	Rib to hold PM ducts, for pole types 1 to 7
ThickMag	Magnet thickness
WidthMag	Total width of all magnet per pole
BarEndExt	One-side damper bar end extended length
RingLength	One-side axial ring length
RingHeight	Radial ring height
LenRegion	Region length
InfoCore	0: core; 1: magnets; 2: damper; 100: region

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/RacetrackSlotCore

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores
Length	Core Length
Skew	Skew angle in core length range
Slots	Number of slots
SlotType	Slot type: 1 to 7
Hs0	Slot opening height
Hs01	Slot closed bridge height
Hs1	Slot wedge height

Hs2	Slot body height
Bs0	Slot opening width
Bs1	Slot wedge maximum width
Bs2	Slot body bottom width, 0 for parallel teeth
Rs	Slot body bottom fillet
FilletType	0: a quarter circle; 1: tangent connection; 2 & 3: arc bottom
Poles	Number of poles
WidYoke	Racetrack width for 2/4-pole cores
MidSlots	Number of middle-size slots for 2/4-pole cores
MidHs2	Middle-size slot body height for 2/4-pole cores
MidBs2	Middle-size slot body bottom width for 2/4-pole cores
SmlSlots	Number of small-size slots for 2/4-pole cores
SmlHs2	Small-size slot body height for 2/4-pole cores
SmlBs2	Small-size slot body bottom width for 2/4-pole cores
LenRegion	Region length
InfoCore	0: core; 100: region

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/SalientPoleCore

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores
Length	Core Length
Skew	Skew angle in core length range
Slots	Number of damper slots per pole
SlotType	Slot type: 1 to 4
Hs0	Slot opening height
Hs01	Slot closed bridge height
Hs1	Slot wedge height
Hs2	Slot body height
Bs0	Slot opening width

Bs1	Slot wedge maximum width
Bs2	Slot body bottom width, 0 for parallel teeth
Rs	Slot body bottom fillet
FilletType	0: a quarter circle; 1: tangent connection; 2 & 3: arc bottom
SlotPitch	Slot pitch in mechanical degrees, refer to offset point
CenterPitch	Center slot pitch in mechanical degrees, refer to offset point
Poles	Number of poles
WidthShoe	Pole shoe width
HeightShoe	Pole shoe height (maximum)
WidthBody	Pole body width
HeightBody	Pole body height
AirGap2	Second air gap length
Offset	Pole arc offset
Off2_x	The second pole arc offset perpendicular to the pole-center line
Off2_y	The second pole arc offset parallel with the pole-center line
CoilEndExt	One-side coil end extended length
EndRingType	0: whole press board; 1: pole press board; 2: pole ring; 3: whole ring
BarEndExt	One-side damper bar end extended length, for types 2 & 3 only
RingLength	One-side axial ring length, or conductor press board thickness
RingHeight	Radial ring height, for types 2 & 3 only
LenRegion	Region length
InfoCore	0: core; 1: core & coils; 2: coil; 3:damper; 4:terminal1; 5: terminal2;

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/SlotCore

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores
Length	Core Length
Skew	Skew angle in core length range
Slots	Number of slots

SlotType	Slot type: 1 to 6
Hs0	Slot opening height
Hs01	Slot closed bridge height
Hs1	Slot wedge height
Hs2	Slot body height
Bs0	Slot opening width
Bs1	Slot wedge maximum width
Bs2	Slot body bottom width, 0 for parallel teeth
Rs	Slot body bottom fillet
FilletType	0: a quarter circle; 1: tangent connection; 2 & 3: arc bottom
HalfSlot	0 for symmetric slot, 1 for half slot
LenRegion	Region length
InfoCore	0: core; 100: region

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/SquirrelCage

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores
Length	Core Length
Skew	Skew angle in core length range
Slots	Number of slots
SlotType	Slot type: 1 to 4
Hs0	Slot opening height
Hs01	Slot closed bridge height
Hs1	Slot wedge height
Hs2	Slot body height
Bs0	Slot opening width
Bs1	Slot wedge maximum width
Bs2	Slot body bottom width, 0 for parallel teeth
Rs	Slot body bottom fillet

FilletType	0: a quarter circle; 1: tangent connection; 2 & 3: arc bottom
HalfSlot	0 for symmetric slot, 1 for half slot
BarEndExt	One-side bar end extended length
RingLength	One-side axial ring length
RingHeight	Radial ring height
RingDiaGap	Ring diameter on gap side
CastRotor	0:insert-bar; 1: cast-rotor
LenRegion	Region length
InfoCore	0: bars & rings; 1: bars; 2: rings; 100: region

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/SRMCore

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores
Length	Core Length
Poles	Number of poles
ThkYoke	Toke thickness
Embrace	Pole embrace (the ratio of pole arc to pole pitch)
EndExt	Coil one-side end extended length
LenRegion	Region length
InfoCore	0: core; 1: core & coils; 2: coil; 3: terminal1; 4:terminal2; 100: region

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/SynRMCore

Property	Description

DiaGap	Core diameter on gap side, or outer diameter
DiaYoke	Core diameter on yoke side, or inner diameter
Length	Core Length
Poles	Number of poles
PoleType	1: ALA; 2: arc; 3: hyperbolic; 4: hyperbolic line
Barriers	Barriers per pole, for PoleType 2 & 3 only
H	Bridge thickness, for PoleType 2 & 3 only
W	Rib width, for PoleType 2 & 3 only
R	Barrier fillet radius, for PoleType 2 & 3 only
R0	Radius of the bottom barrier arch, for PoleType 2 & 3 only
Rb	Barrier bottom minimum radius
Y0	Yoke bottom thickness
B0	Barrier bottom thickness, for PoleType 2 & 3 only
LenRegion	Region length
InfoCore	0: core; 1: one barrier; 100: region

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/TransCoil

Property	Description
DistLeg	Leg center to center distance
CoilType	Coil type: 1 for solenoid coil; 2 for pancake coil
WidthIn	Coil width between two inner sides
DepthIn	Coil depth between two inner ends
RadiusIn	Coil inner fillet radius
ThickCoil	Coil thickness of one side
HighCoil	Coil Height
Layers	Number of layers
GapLayer	Gap between two layers
InfoCore	0: all coils; 1: one coil only

Related Topics

Parameters for RMxprt User Defined Primitives
Specifying UDP Parameters

RMxprt/TransCore

Property	Description
DiaLeg	Outer diameter of leg cross-section
DistLeg	Leg center to center distance
DistYoke	Yoke center to center distance
Stages	Number of stages of leg cross-section
ThickCore	Core thickness, only used for Stages=1
WidthYoke	Yoke width, =0 for same cross-section as leg's
InfoCore	0: whole core; 1: legs only; 2: yokes only

Related Topics

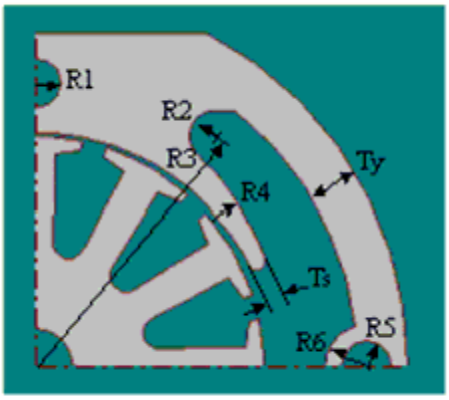
Parameters for RMxprt User Defined Primitives
Specifying UDP Parameters

RMxprt/UnivMCore

See graphic below for definitions.

Property	Description
DiaGap	Core diameter on gap side, or outer diameter
DiaYoke	Core diameter on yoke side, or inner diameter
Length	Core Length
Skew	Skew angle in core length range
Poles	Number of poles
PoleType	Pole type: 1 to 2
WidthCore	Overall width of a racetrack core
Embrace	Pole embrace (the ratio of pole arc to pole pitch)
Offset	Pole arc offset
WidthPole	Minimum width at pole root

Ty	Yoke thickness
Ts	Shoe-tip thickness
R1	Radius of the screw holes in pole center
R2	Radius of the side fillet arc at pole root
R3	Radius of the side fillet arc center layout circle
R4	Radius of the shoe connecting arc, 0 for auto-design
R5	Inner radius of the screw holes in between two poles
R6	Outer radius of the screw holes in between two poles
EndExt	Coil one-side end extended length
LenRegion	Region length
InfoCore	0: core; 1: coils; 2: one coil; 3: terminal1; 4: terminal2: 100: region



Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/VentSlotCore

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores
Length	Core Length
Skew	Skew angle in core length range

Slots	Number of slots
SlotType	Slot type: 1 to 7
Hs0	Slot opening height
Hs01	Slot closed bridge height
Hs1	Slot wedge height
Hs2	Slot body height
Bs0	Slot opening width
Bs1	Slot wedge maximum width
Bs2	Slot body bottom width, 0 for parallel teeth
Rs	Slot body bottom fillet
FilletType	0: a quarter circle; 1: tangent connection; 2 & 3: arc bottom
HalfSlot	0 for symmetric slot, 1 for half slot
VentHoles	Number of axial vent holes per row
HoleDiaIn	Diameter of inner vent holes
HoleDiaOut	Diameter of outer vent holes
HoleLocIn	Diameter of inner vent hole center layout circle
HoleLocOut	Diameter of outer vent hole center layout circle
VentDucts	Number of radial vent ducts
DuctWidth	Axial width of radial vent ducts
DuctPitch	Center-to-center distance between two adjacent ducts
LenRegion	Region length
InfoCore	0: core; 100: region

Related Topics

Parameters for RMxprt User Defined Primitives

Specifying UDP Parameters

RMxprt/WaveCoil

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores
Length	Core Length
Skew	Skew angle in core length range
Slots	Number of slots

SlotType	Slot type: 1 to 7
Hs0	Slot opening height
Hs1	Slot wedge height
Hs2	Slot body height
Bs0	Slot opening width
Bs1	Slot wedge maximum width
Bs2	Slot body bottom width, 0 for parallel teeth
Rs	Slot body bottom fillet
FilletType	0: a quarter circle; 1: tangent connection
Layers	Number of winding layers
CoilPitch	Coil pitch measured in slots
EndExt	One-side end extended length
SpanExt	Axial length of end span; 0 for no span
SegAngle	Angle per segment of end span; 0 for true-surface end span
PolePitch	Pole pair pitch measured in slots
LenRegion	Region Length
InfoCoil	0: winding; 1: one coil; 100: region

Related Topics

Parameters for RMxprt User Defined Primitives

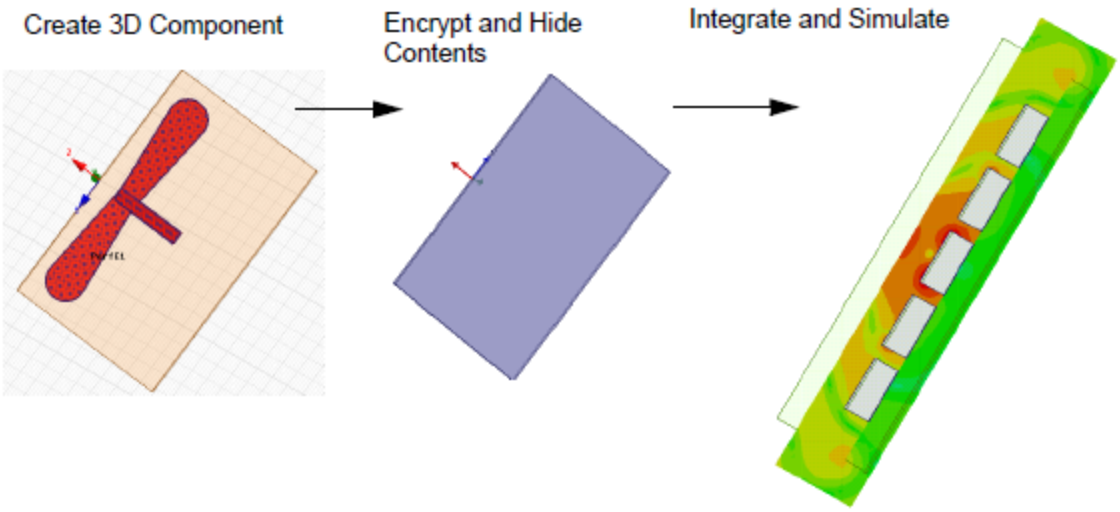
Specifying UDP Parameters

3D Component Library

The 3D Modeler lets you access predefined 3D component libraries and create 3D components with:

- Geometry – objects, parameters, and coordinate systems.
- Design data, boundaries, excitations and mesh operations.
- Material assignment and definition for components are separate from project material and other component materials. They do not appear in the Project Tree materials list under Definitions. Component materials cannot be edited in the project in which they are inserted.
- Component parameters, including variables used by geometries, design data, and material properties. These are displayed separately on the Parameters tab.
- Optional logo or icon image to display with components in the modeler window.

- Optional encryption with password definition, or internal key as a password.



The ability to create components from models lets you easily share and reuse components. You can add components to a library and share components with other users. Once you insert a component into the target design, you can directly manipulate whole components using Arrange operations such as **Move**, **Rotate** and **Mirror or** Duplicate operations like **Duplicate Along Line**, **Duplicate Around Axis** and **Duplicate Mirror**.

Note	3D Component Libraries can contain components created using the **Create 3D Component** menu command in both Maxwell 2D and Maxwell 3D projects. 3D Components created in Maxwell 2D projects cannot be used in Maxwell 3D projects, and vice-versa.

Instances of an inserted component are grouped by definition in the model history. Multiple instances of identical components permit mesh reuse.

Note	Except where noted, the following sections apply to component libraries and components created in both Maxwell 2D and Maxwell 3D projects.

The documentation includes the following sections:

Creating a 3D Component from an Existing Model

Replace Selection with 3D Component

Inserting a 3D Component into a Design

Using the Component Libraries Window

Viewing Properties of 3D Components

3D Components in the Project Manager Window

3D Components in the History Tree

Editing 3D Component Properties

Editing 3D Component Definitions

Updating 3D Component Definitions

Human Body Exteriors

Qi Wireless Power Transfer System

3D Component Agents

Related Topics

Fit All Objects in a View Window.

Fit Selected objects in a View Window.

Showing Only Selected Objects in All or Active Views

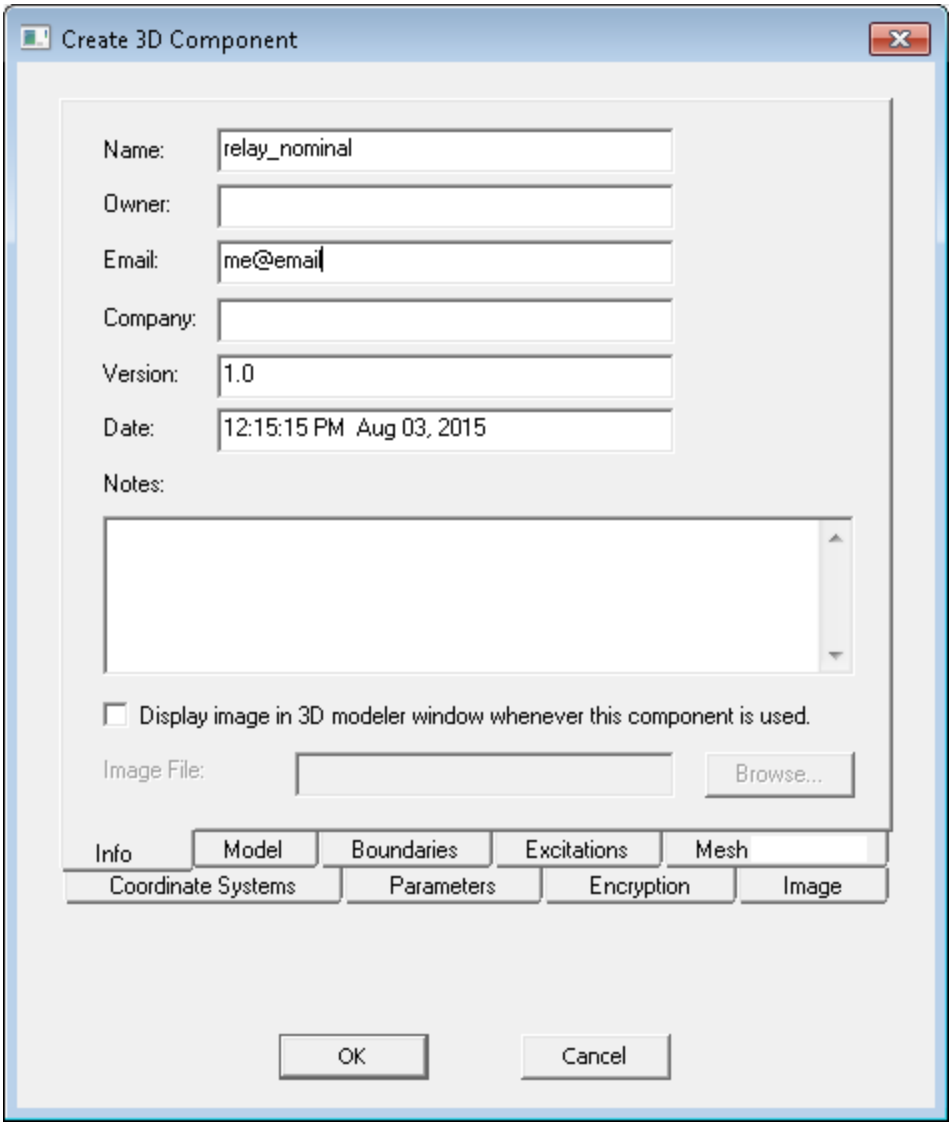
Hiding Objects from View

Creating a 3D Component from an Existing Model

To create a 3D component, select an existing object or model, right-click, and select **Create 3D Component**. You can also click **Draw>3D Component Library>Create 3D Component.**

This opens the **Create 3D Component** dialog box. The **Model** tab lists all of objects selected before you invoked **Create 3D Component**.

Note	The Create 3D Component menu command can be used to create components in both Maxwell 2D and Maxwell 3D projects. 3D components created in Maxwell 2D cannot be used in Maxwell 3D projects, and vice-versa.



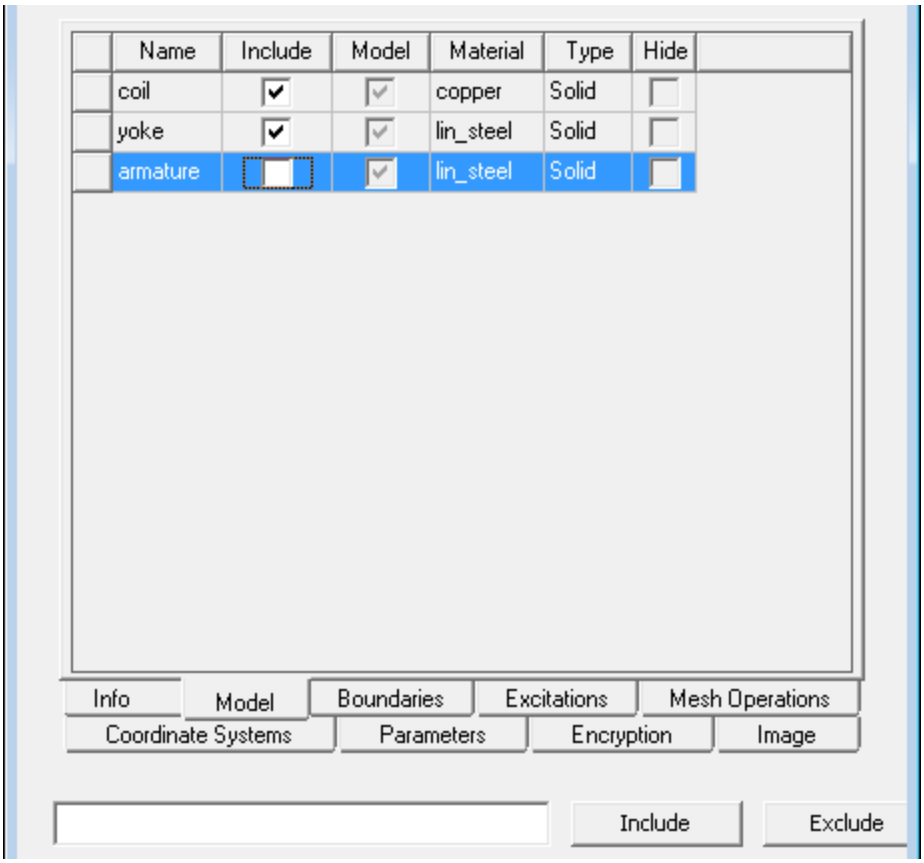
The tabs let you view the component features organized according to geometry, design data, and settings:

- Info tab
- Model tab
- Boundaries tab
- Excitations tab
- Mesh tab
- Coordinate Systems tab
- Parameters tab
- Encryption tab
- Licensing tab
- Image tab

Note	3D components selected from the SysLib libraries include tabs only for Parameters, Image, and Info.

Include or Exclude Feature from Component

You can view the various tabs and select which features to **Include** or **Exclude** in creating the component. You can use the check boxes in the Include column for this purpose. You can also use the text field. Type a feature Name which you can then use the command buttons to Include or Exclude.



Object selection for **Include** or **Exclude** impacts selection in other tabs. For example, if you exclude an object from the component creation, the design data, parameters, and coordinate systems corresponding to that object are also excluded.

Sorting Columns Using the Headers

The features listed for each tab include columns for the Name, check boxes for whether to include, and properties. You can click on the column headers to sort by Name, Model, or other feature listed for each tab. Click the column header to select the column to sort. Click again to invert the column.

You can sort lists by using all columns except Include.

Notes on Design Data

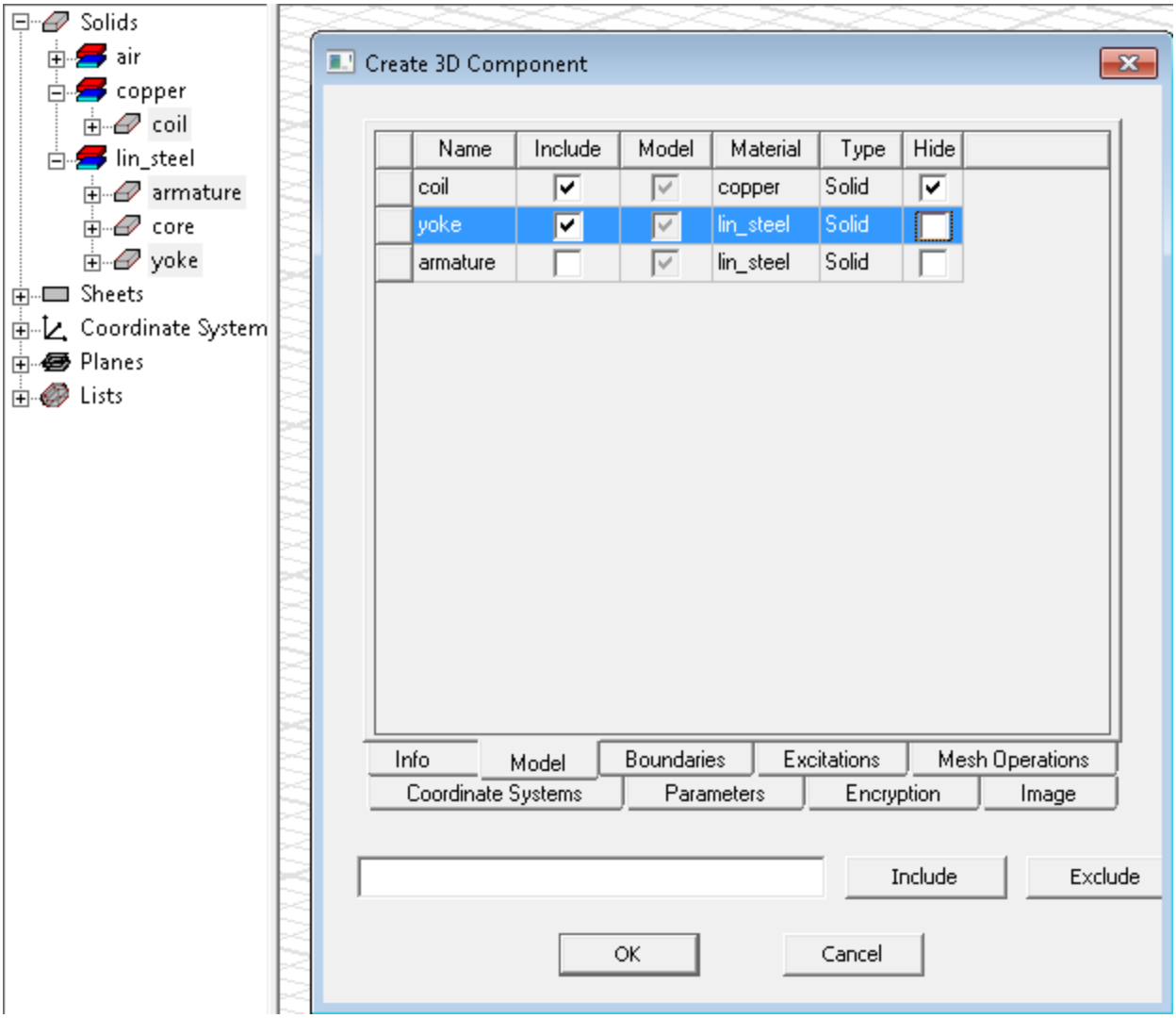
- DC thickness on selected objects is always included in components. It is not listed in **Boundaries** tab.
- The **Create 3D Component** dialog does not list design data without an assignment (for example, a winding in Maxwell).
- The parent of any included boundary/excitation is included, as long as the parent does not require assignment (for example, a winding in Maxwell).
- Design settings like material overrides are not included.

Once you have made the Include and Exclude settings, and have specified the Image and Info you click **OK** to Save 3D Component File. When you click OK, the component is validated for everything that is included in the component. For example, you cannot include boundaries if the object on which the boundaries have been created is not included. After validation, a dialog for saving the component to a location appears.

Create 3D Component: Model Tab

The **Model** tab lists all of objects selected before you invoked **Create 3D Component**. These correspond to the selected objects listed in the history tree.

- Use the check boxes or text field to Include or Exclude objects for the component.
- Material and Type columns are read only.
- For Encrypted components, you can also chose which objects to hide. The hide column is only enabled if you have the Hide Contained option selected on Encryption tab.

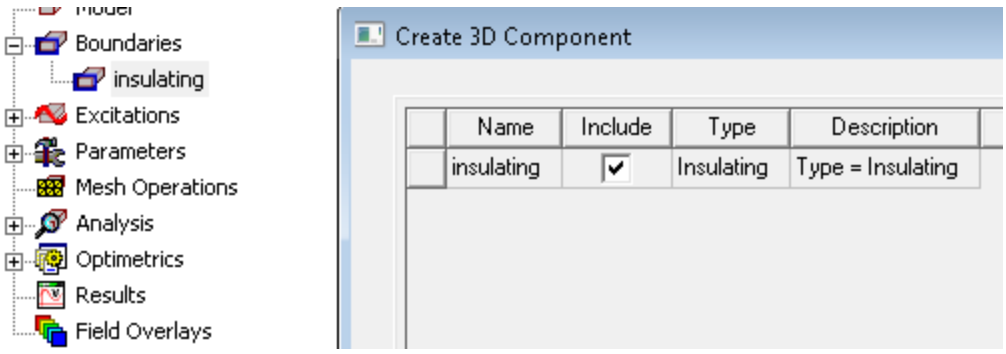


Create 3D Component: Boundaries Tab

The list is populated with all the boundaries for the selected objects.

- Use the **Include** check boxes to include (selected) or exclude (cleared) individual boundaries for the 3D component.

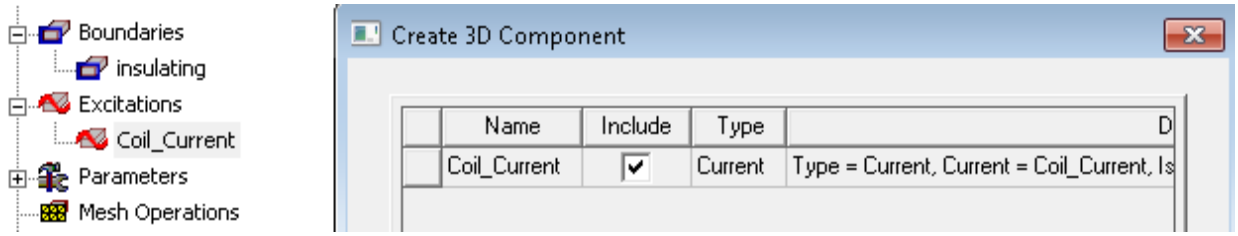
  Alternatively, you can type a boundary Name into the text box and click the **Include** or **Exclude** button. Wildcards (? and *) are supported when you use this method, so you can change the inclusion state of multiple boundaries with similar names in a single operation.
- Type and Description columns are read only.



Create 3D Component: Excitations Tab

The list is populated with all the excitations for the selected objects.

- Use the **Include** check box to include excitations in the 3D component (selected), or to Exclude them from the component (cleared). You can also use the edit box and **Include**/**Exclude** buttons to filter excitations.
- *Type* and *Description* columns are read only.

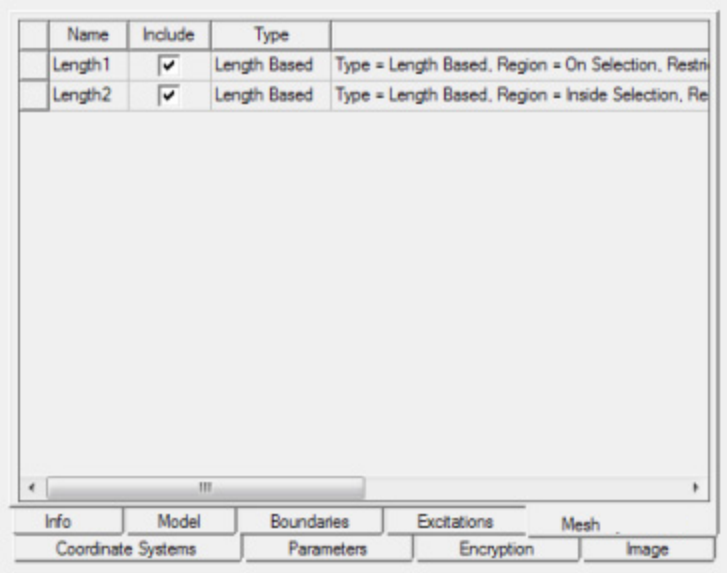


Create 3D Component: Mesh Tab

The list is populated with all the mesh operations for the selected objects.

- Use the Include check boxes to include (selected) or exclude (cleared) individual mesh operations.

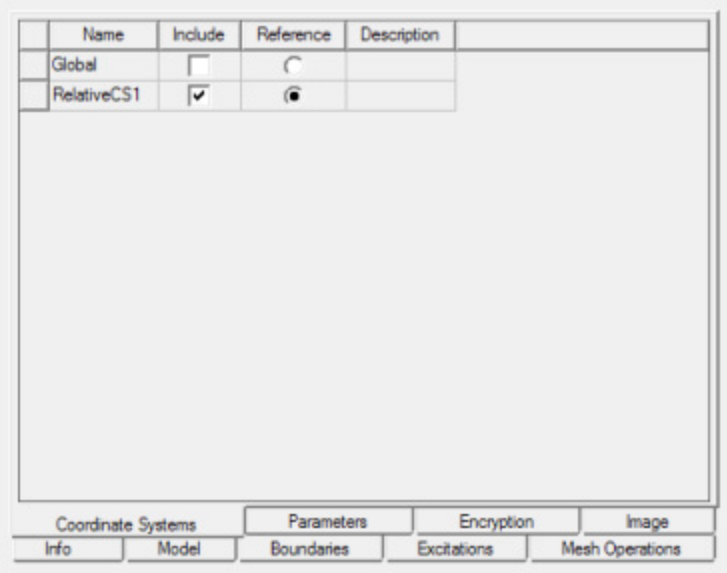
  Alternatively, you can type a mesh operation Name into the text box and click the **Include** or **Exclude** button. Wildcards (? and *) are supported when you use this method, so you can change the inclusion state of multiple mesh operations with similar names in a single operation.
- *Type* and *Description* columns are read only



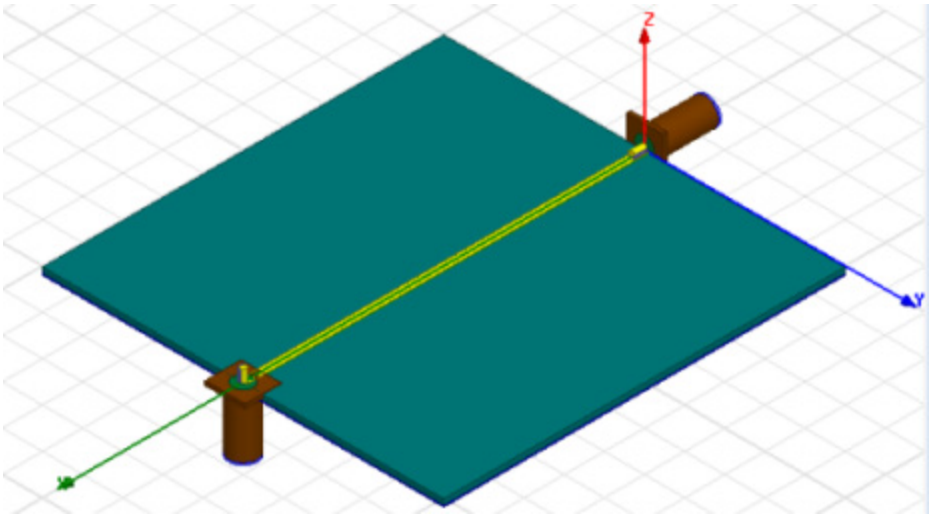
Create 3D Component: Coordinate Systems Tab

The list is populated with all the Coordinate Systems for the selected objects. By default, only the Coordinate Systems used to define the object orientations are included. You can include Coordinate Systems created on specific model parts.

- Use the checkbox to Include or Exclude coordinate systems. You can also use the text field and **Include**/**Exclude** buttons. Global CS cannot be included in the component.
- You can select any of the CS as a component reference.
- By default, the current working CS is the component reference CS.
- Reference CS must be included in the model (except for global).



The coordinate Reference system that you specify affects the orientation of the component upon insertion.

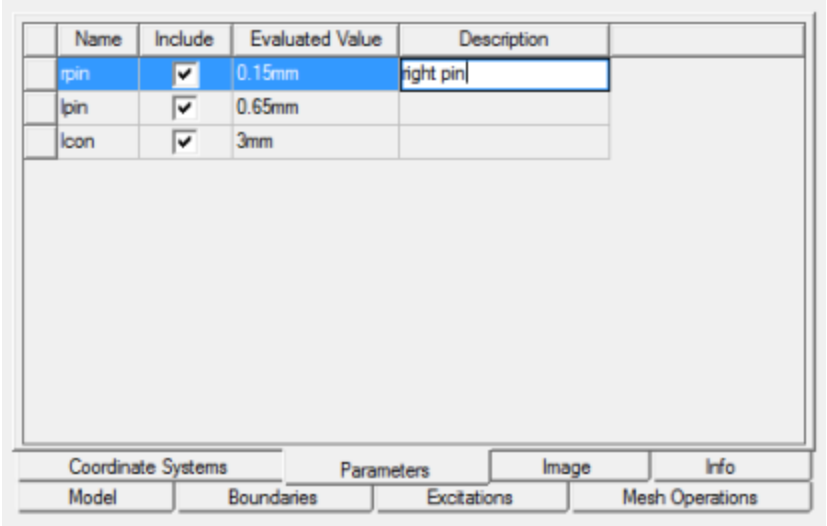


Create 3D Component: Parameters Tab

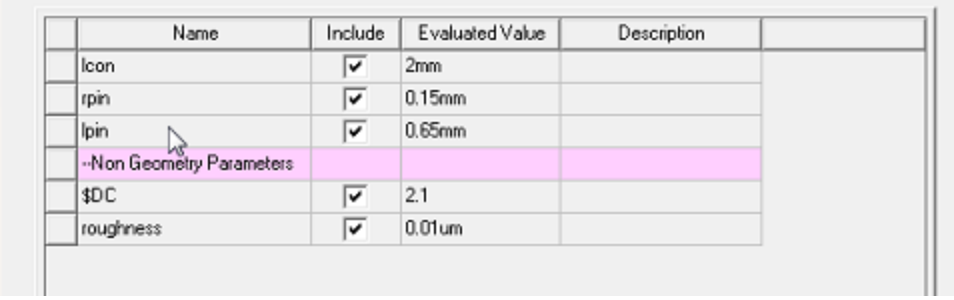
Component properties can be parameterized by assigning variables. The parameters list is populated with all the variables used by objects (and Coordinate Systems, design data, and material properties) included in the component.

- Use the **Include** check box to include parameters in the 3D component (selected), or to Exclude them from the component (cleared).

  Alternatively, you can type a parameter Name into the text box and click the **Include** or **Exclude** button. Wildcards (? and *) are supported when you use this method, so you can change the inclusion state of multiple parameters with similar names in a single operation.
- You can fill in the description field, if desired.



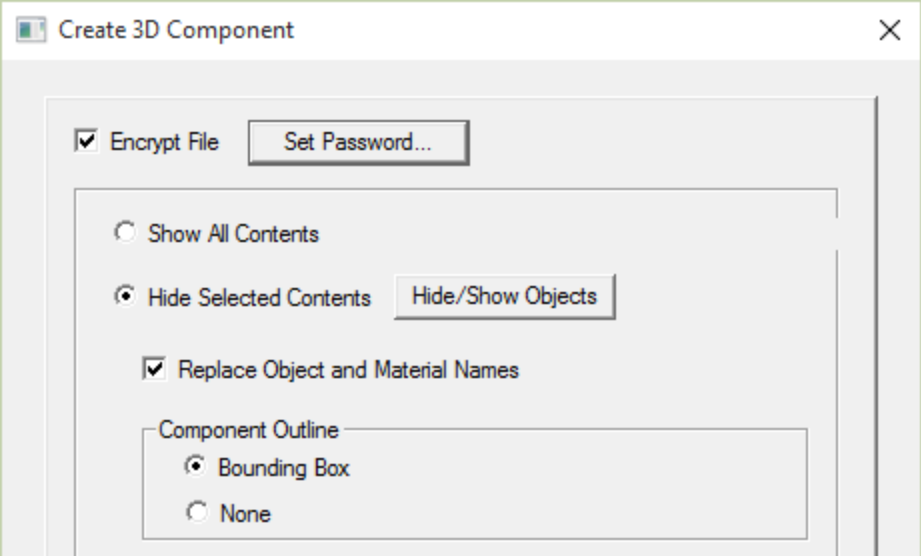
If you create non-geometry variables materials or boundaries, these are listed separately:



If you exclude any geometry, boundary or material associated with a variable, the associated variable is automatically excluded on the **Parameters** tab.

Create 3D Component: Encryption Tab

The **Encryption** tab includes a checkbox for you to define a user access password to use the component.



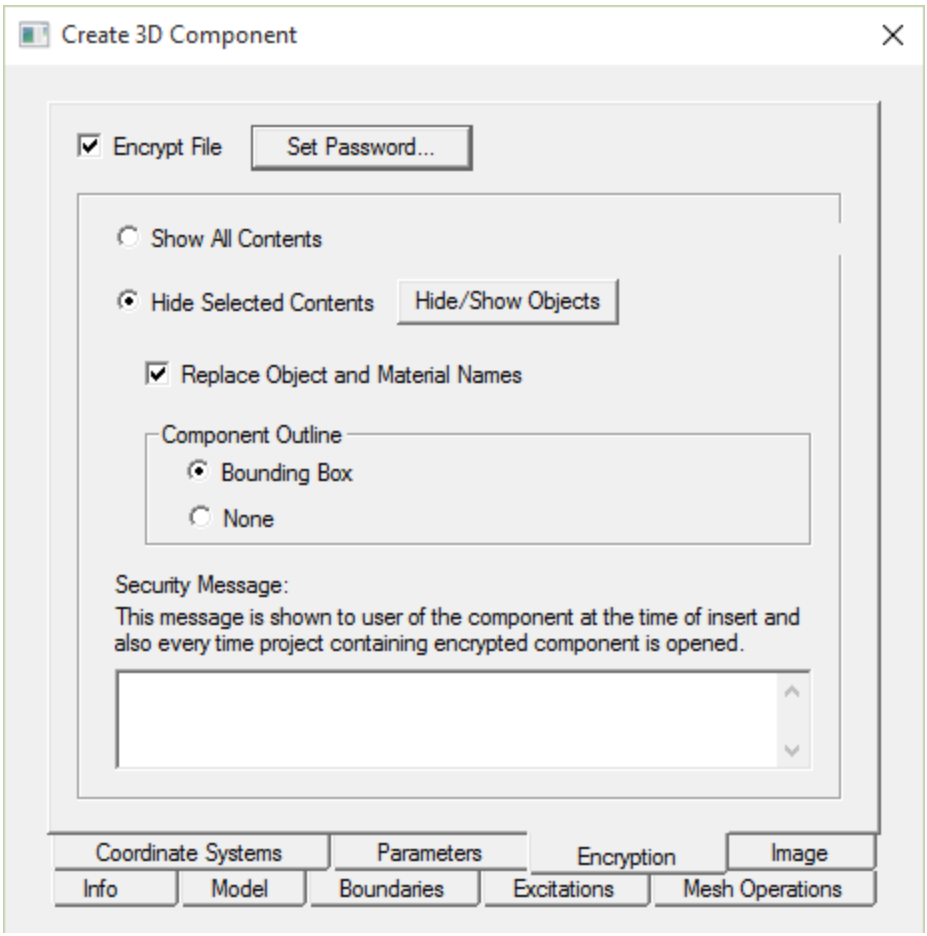
Encryption is a common technique for encoding information in a manner that is very difficult for unauthorized users to decode. When this option is enabled for 3D Components, the contents of the component file are encrypted to protect them from external viewing or editing.

- 3D Components use the Advanced Encryption Standard (AES) with a 256-bit key and password-based key derivation.
- If the encrypted component is created with an **Internal Key**, it can be used without additional authorization.
- If the encrypted component is created with a **User Password**, that password must be entered when the component is used in a design.

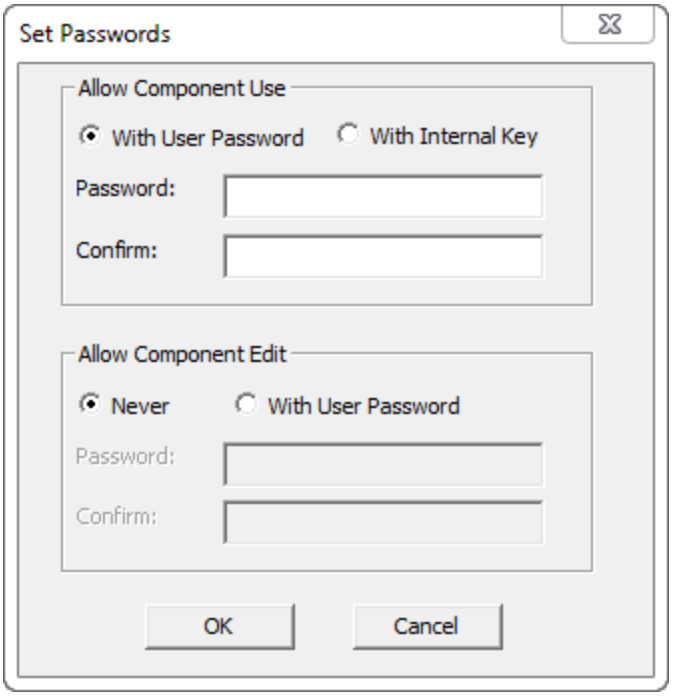
- It is not possible to edit the definition of an encrypted component within the Electronics Desktop.
- If the **Show All Contents** option is selected, the geometry and other contents will be shown when the component is used. Geometry export and links will operate as for general geometry.
- The **Hide Selected Contents** option can be used to further protect access to an encrypted component's details. When this option is selected, hidden contents will not appear in the Electronics Desktop user interface.
    - All geometry will be hidden, except for objects that the user specifies as visible during component creation.
    - Hidden geometry will not be visible in the model window or included in the history tree.
    - Geometry export and links will not be allowed.
    - Materials, boundaries, and other setup data will also be hidden, with limited exceptions such as excitation settings that must be accessed in order to control the simulation and post processing of the design.
    - For components with hidden contents, all geometry will be excluded from field calculations and from plots of fields and meshes. Plots in surrounding volumes **will extend to, and can include, the outer surfaces of the component**, but not the containing volume.
    - **Note**: Such plots may indirectly reveal component content by providing a view of the surfaces where the plots terminate. To prevent such exposure, include additional modeled objects that surround the hidden contents of the component and have material of the surrounding material environment (e.g. air, vacuum).
- When using the **Hide Selected Contents** option, the names of objects, materials, boundaries, and other setup data may appear in progress updates, messages, and other informational text. The **Replace Object and Material Names** option can be used to automatically convert object and material names to generic names in the exported component. Other sensitive names should be changed before exporting.
- When a project contains one or more encrypted 3D Components, the project file is encrypted in order to protect its contents. This encryption uses an internal key, so that you can open the project file without entering a password.

Note	Note that there are some additional considerations with respect to projects using encrypted components. The component itself will always be stored in encrypted form. When an HFSS or HFSS-IE design contains an encrypted component with hidden contents, it will also protect other project files containing data that could be used to extract component information. As of the R18 release, other design types, such as Maxwell or Q3D, do not support this additional security.

Checking the **Encrypt File** box enables the **Set Password** button, and selection buttons for whether the password is to **Show All Contents**, or to **Hide Selected Contents**. You can also specify a security message. If you provide a message, it is displayed at component insertion, and when the project containing the component is opened.



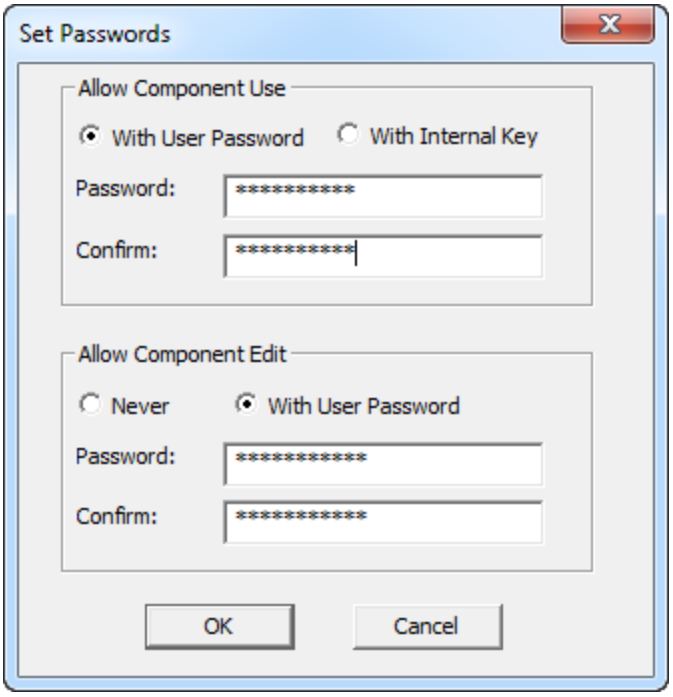
Clicking the **Set Password** button opens a dialog with fields for **Password**, and **Confirm**.This dialog also opens automatically when you check **Encrypt File**.



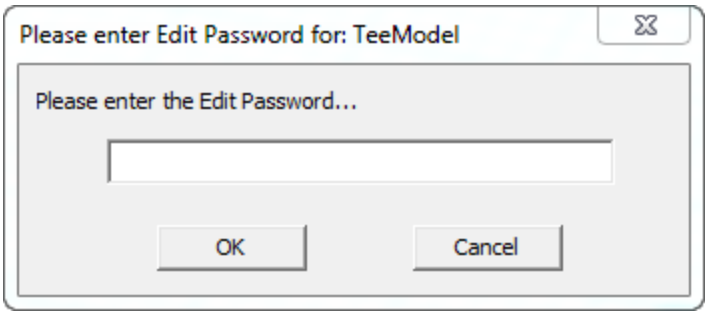
If you specify and OK the fields, the password is set. If you cancel, you will be prompted to provide a password before you save the 3D component.

Selecting **With Internal Key** disables the **Password** and **Confirm** fields. If you select **With Internal Key**, you don't have to specify the password when you use the component.

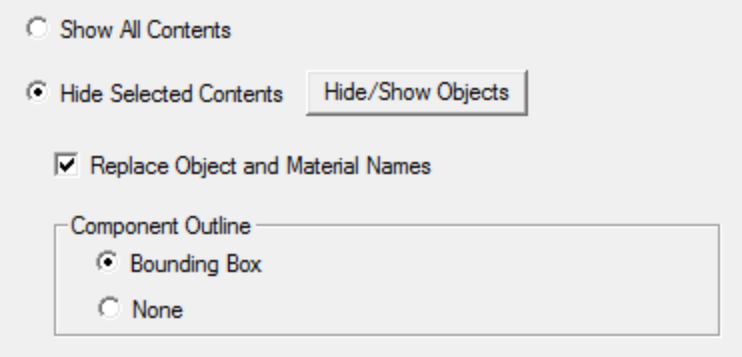
If you change **Allow Component Edit** to **With User Password**, you also specify a **Password** and confirmation. The password for Allow Component Edit does not have to be the same as the password for Allow Component Use. If either Confirm entry does not match the corresponding password, you will be prompted to correct the situation.



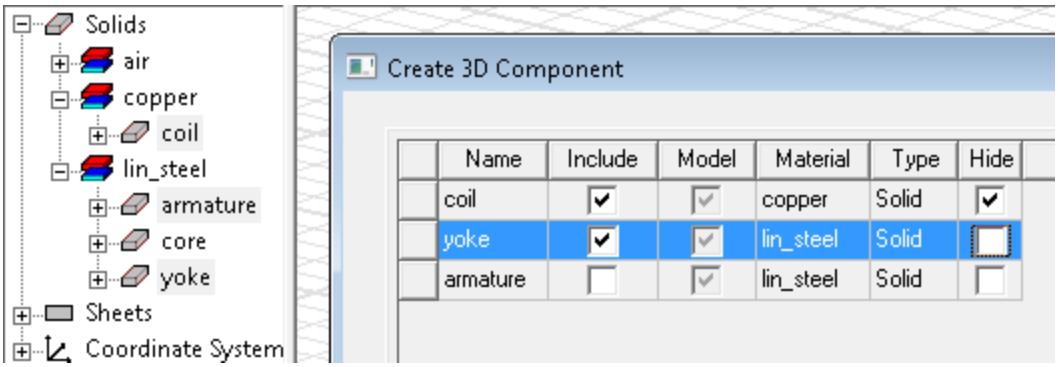
If you then attempt to edit the definition of an encrypted component for which editing is allowed, you will be prompted to enter the Password. You will have three tries to enter the correct password.



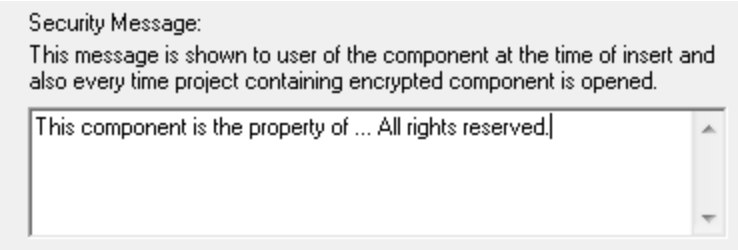
You have a choice regarding whether to **Show All Contents** or whether to **Hide Selected Contents**. If you select **Hide Selected Contents**, you then specify whether to display the component outline as a **Bounding Box** or **None**.



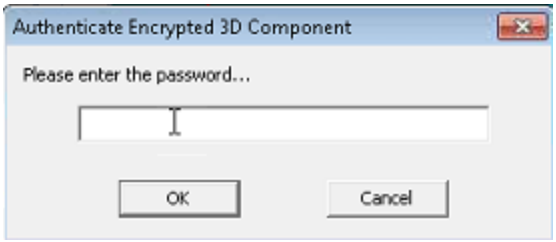
You can also control the visibility per model object by using the Hide check boxes in the Model tab.



You can specify a **Security Message** in the text field. The text you provide displays when you insert a component and when you open a project containing the component.



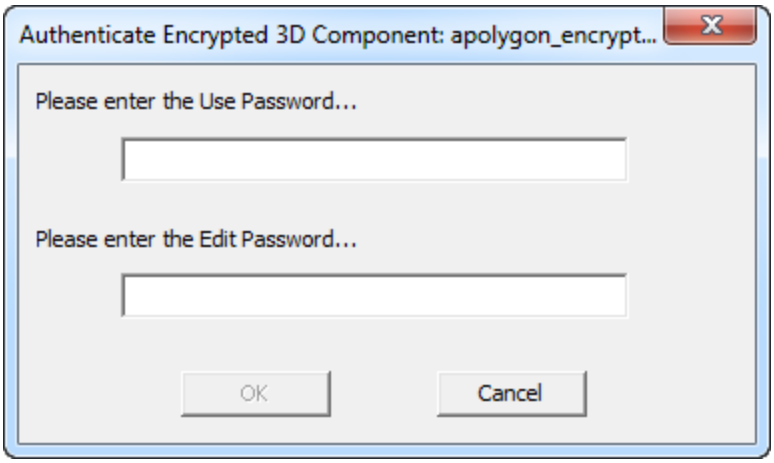
When you insert an encrypted component, you are presented with an Authentication dialog in which you enter a password. A user gets three attempts to enter the correct password.



After authentication, the component is inserted for use. If you have specified a security message, that is displayed.

If you choose to edit the definition for the inserted component, you will be prompted to enter the Edit password.

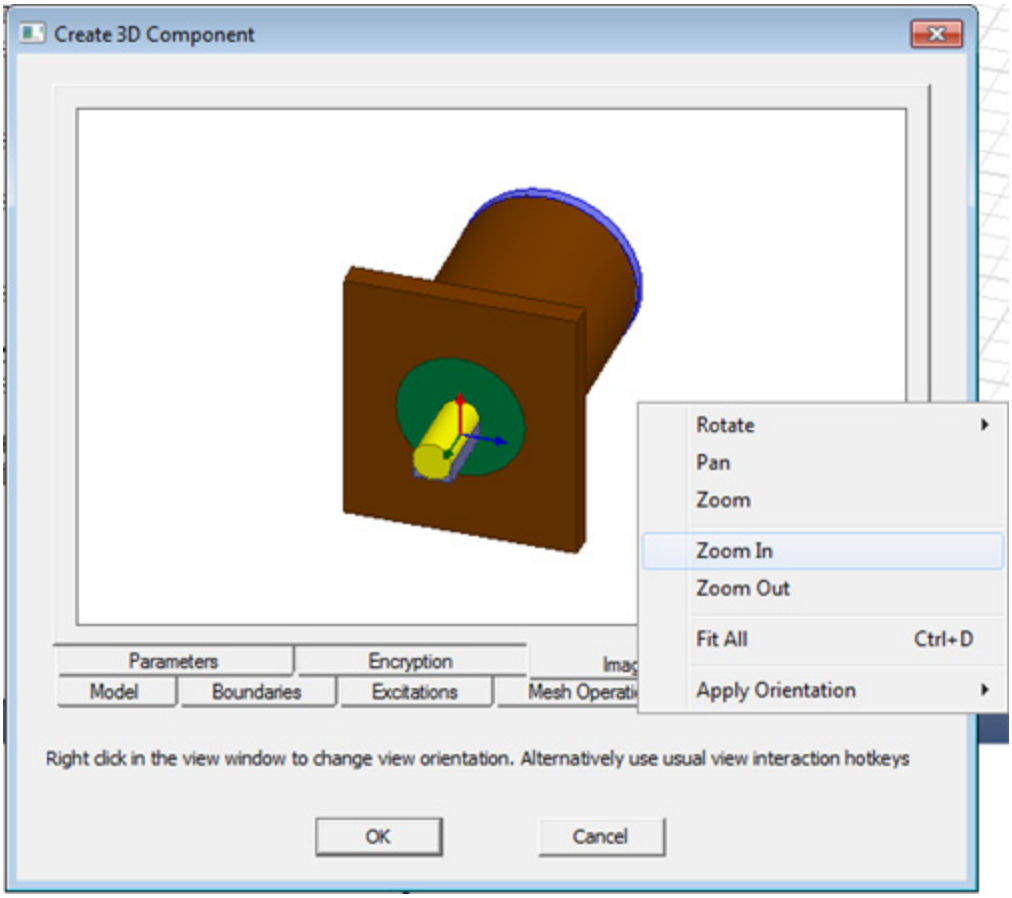
If you decide to open an encrypted but editable 3D Component file using **File>Open**, you are required to enter both the use password (unless encrypted with an internal key) , and the edit password.



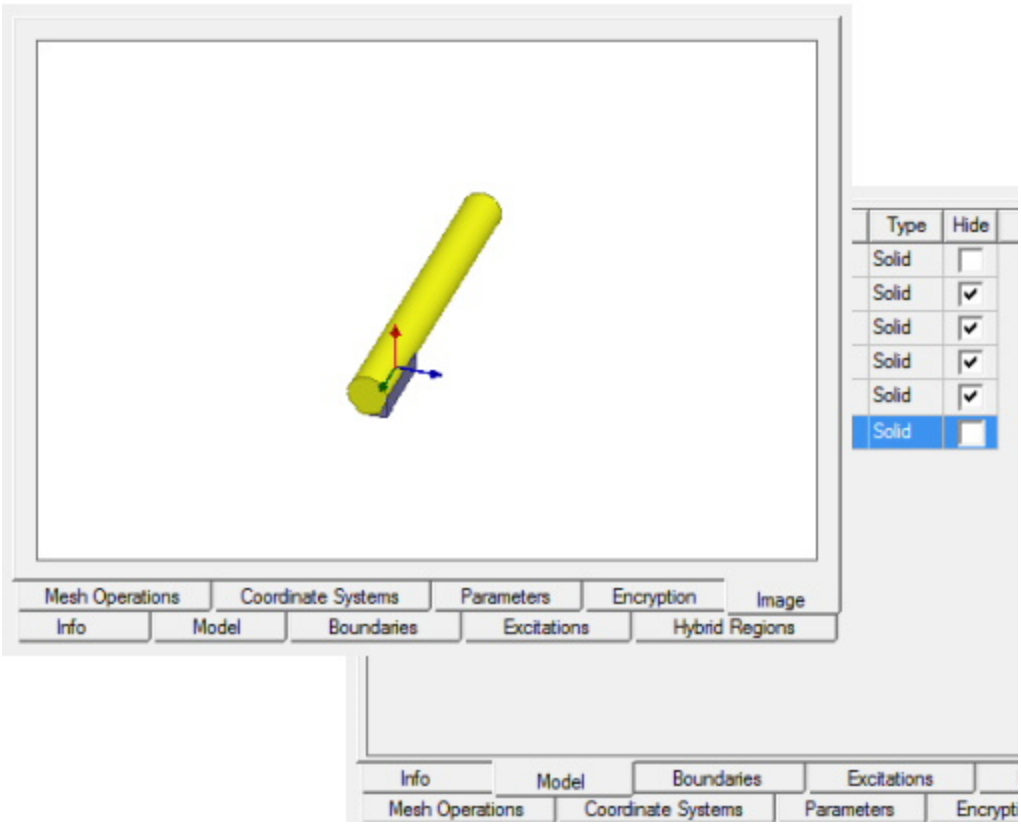
Create 3D Component: Image Tab

The **Image** tab shows the preview image for the component, based on the reference Coordinate System. The image also responds when you **Include** or **Exclude** objects from the **Model** tab.

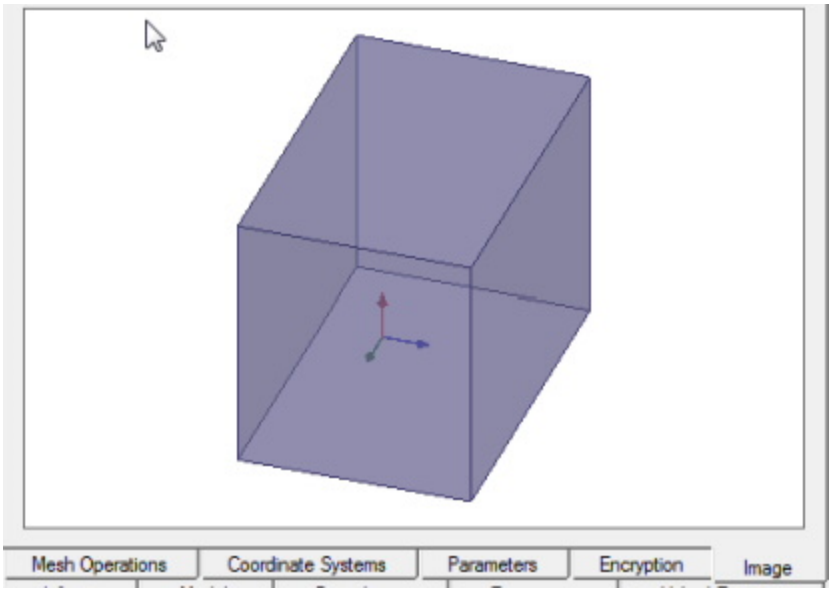
- You can right-click for a menu to change the view of the preview image.
- Changing the preview orientation does not affect the modeler window view.



If you use the Encryption tab and/or Model tab settings to hide/show the model or specified model objects, the Image tab displays accordingly.

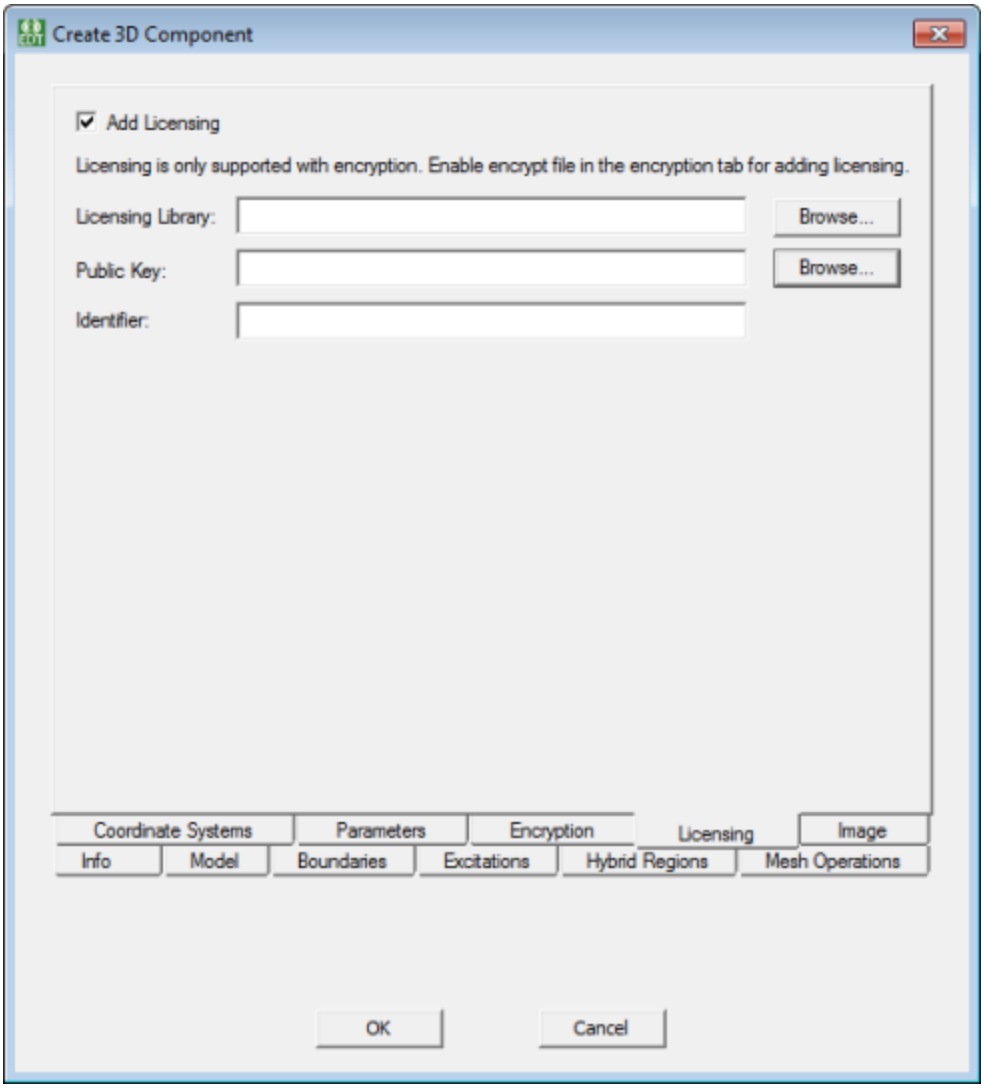


If you use the Encryption tab to display only an object outline, the Image tab displays accordingly.



Create 3D Component Licensing Tab

A 3D Component can be licensed using an external licensing library. This feature is for those who want to license and distribute their 3D Component models. On the **Licensing** tab, you specify a 3D Component Licensing dll, a public key and an optional component identifier. You can license a 3D Component only if it is also encrypted.

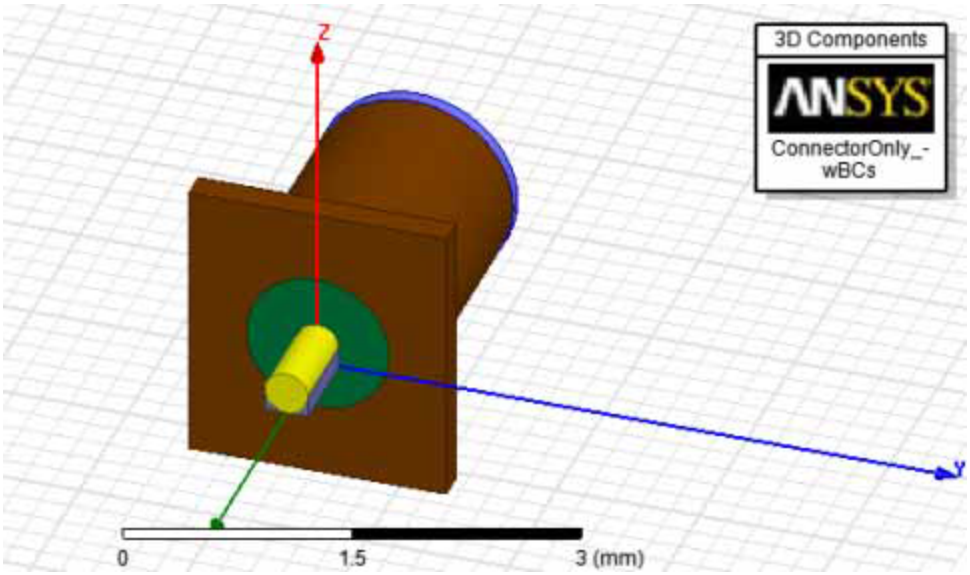


Create 3D Component: Info Tab

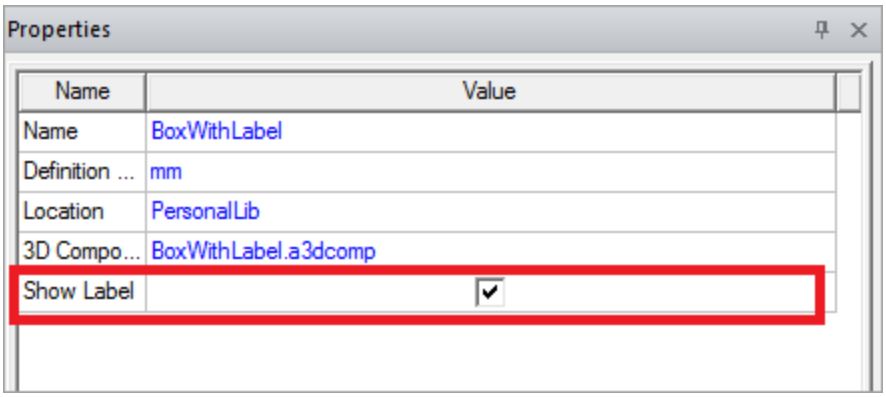
On the **Info** tab you can view and edit component information, as well as add notes. The *Name* in the first field is used when the component is inserted into a design. Other fields include *Owner, email, Company, Company URL, Model Number, Help URL version, Date,* and *Notes*.

You can also included a .bmp format image file to display in the upper right area of the 3D modeler whenever the component is used. The image file typically contains a company logo. Check the

Display image box to enable the **Image File** field, and click **Browse** to open a window to navigate your file system select the image.

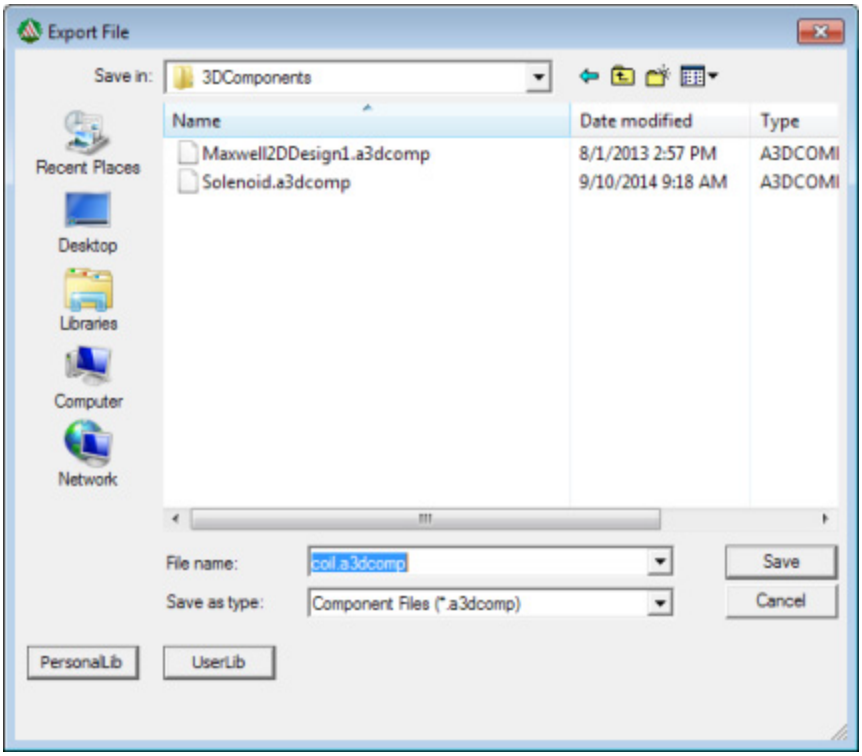


You can move the image in the modeling workspace by clicking the "3D Components" area of the box and dragging it. You can hide the image by editing the 3D component and clearing the **Display image...** check box, or by deselecting the Show Label check box in the Properties window:



Save 3D Component File

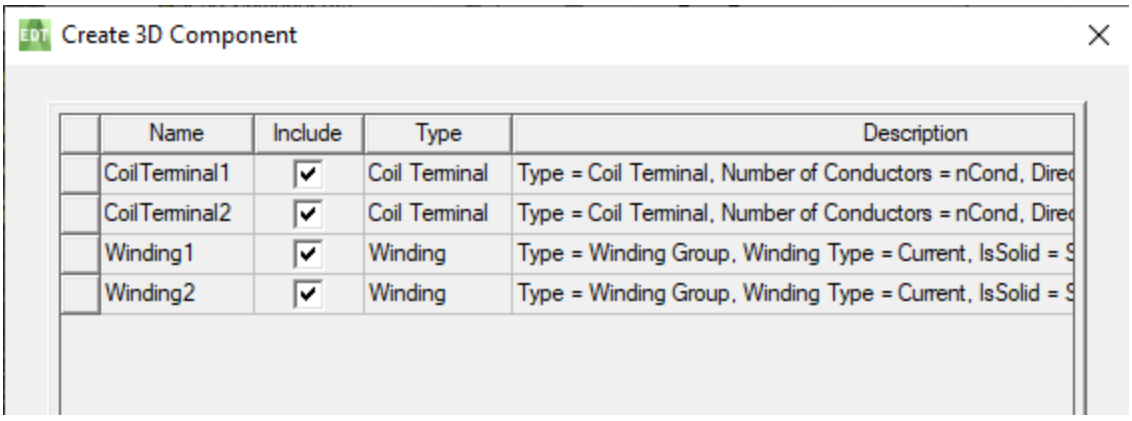
When you **OK** the **Component File** dialog box, an **Export File** dialog displays. By default the Save in field shows the model source directory. Click **PersonalLib** or **UserLib** to display a 3D Components directory.



The default File name is the Component name specified in the **Info tab**.

Maxwell 3D Component with Windings *(2D and 3D Eddy Current and Transient)*

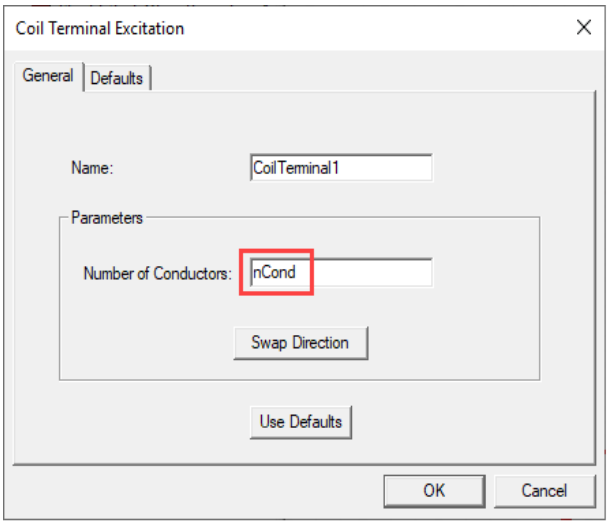
For Maxwell 2D and 3D Eddy current and Transient designs, both voltage and current windings can be included in 3D component definitions on the Excitations tab.

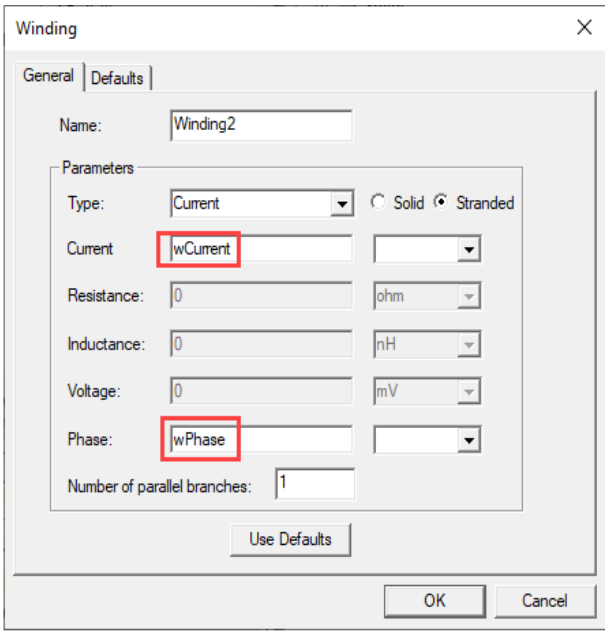


- Windings can only be included in a 3D component when all of its related coil terminals are included in the component as well.

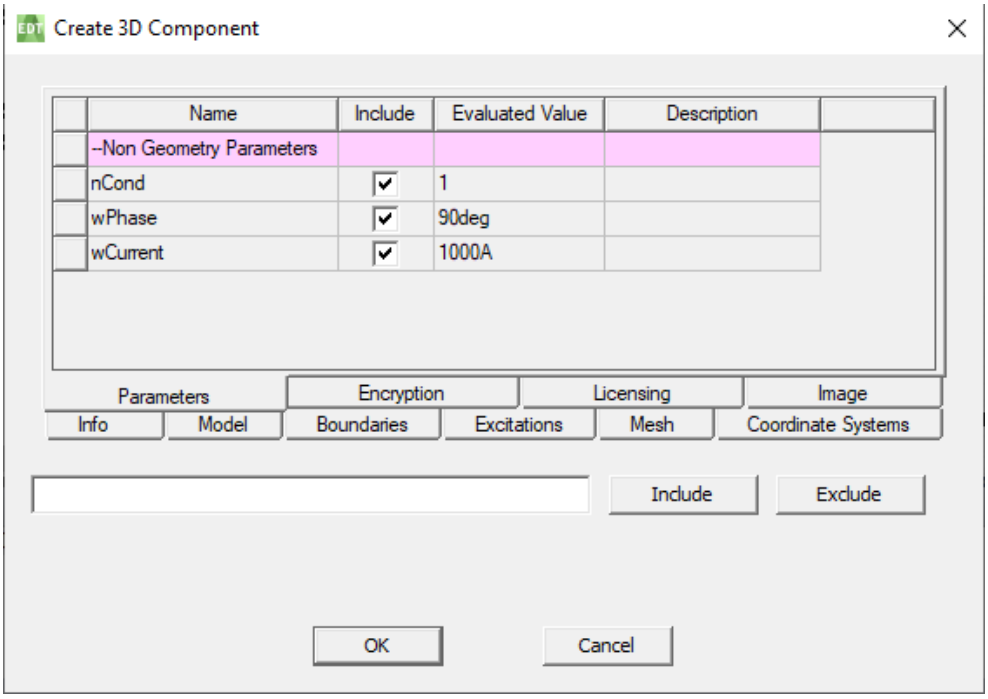
- After inserting a 3D component with windings into a design, only the names of the windings can be edited. Terminals cannot be added to, or removed from, these windings; nor can the windings and associated terminals be removed from the design's Excitations.
- Coil terminals that are assigned to 3D component windings cannot be reassigned to other windings.
- 3D component winding properties (non-geometry parameters) such as: type, resistance, inductance, etc., cannot be edited in a design that includes these 3D components.

  However, any of the winding properties *can* be exposed for editing by assigning design variables to these properties prior to creating the 3D component.

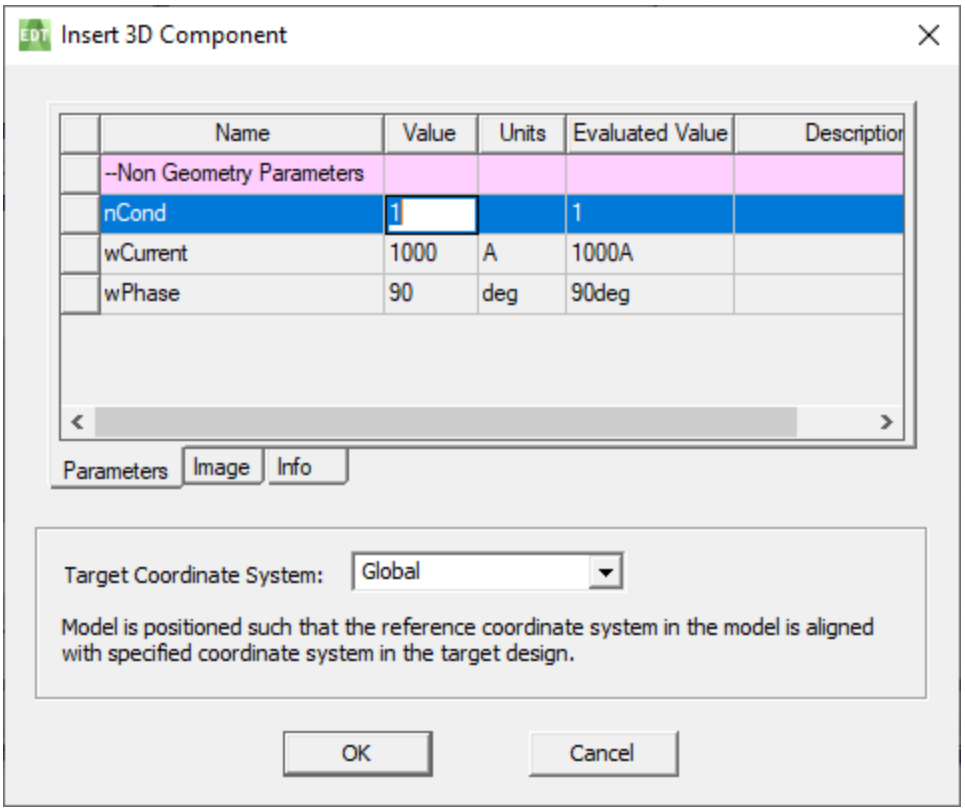




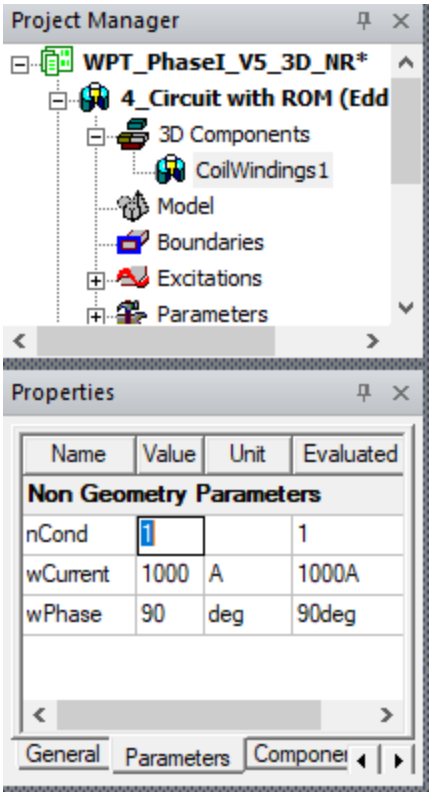
The non-geometry parameters in the example shown above can then be included when creating the 3D component.



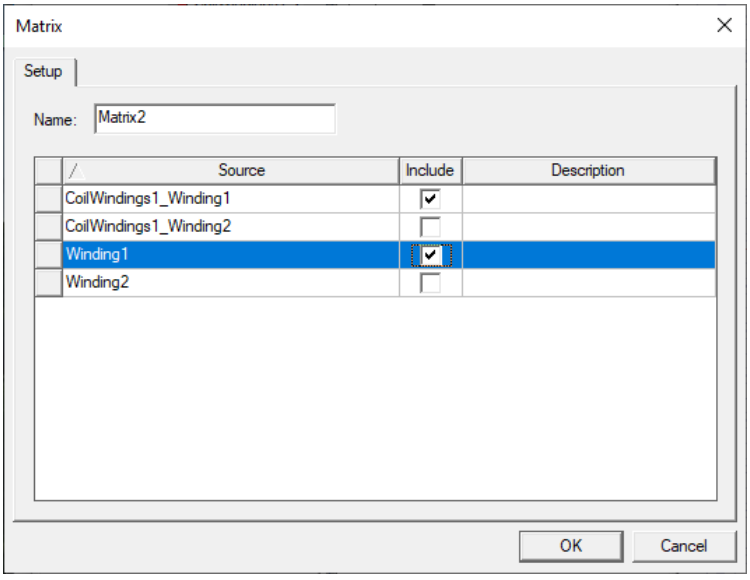
When inserting the new 3D component in a design, parameter values can then be specified.

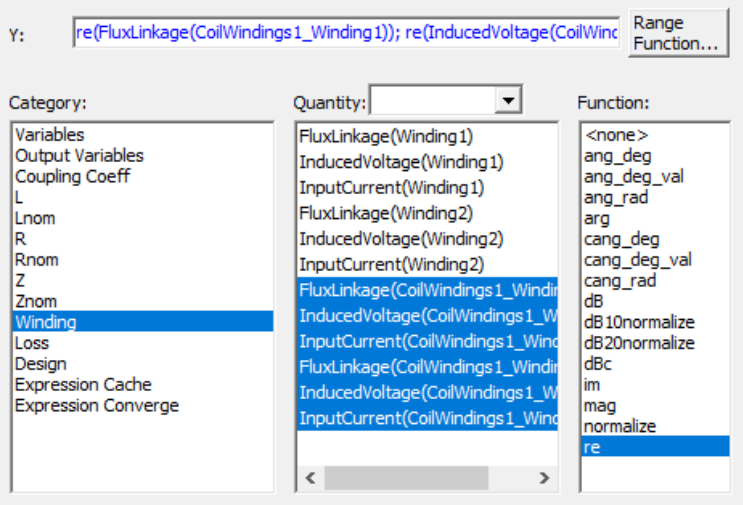


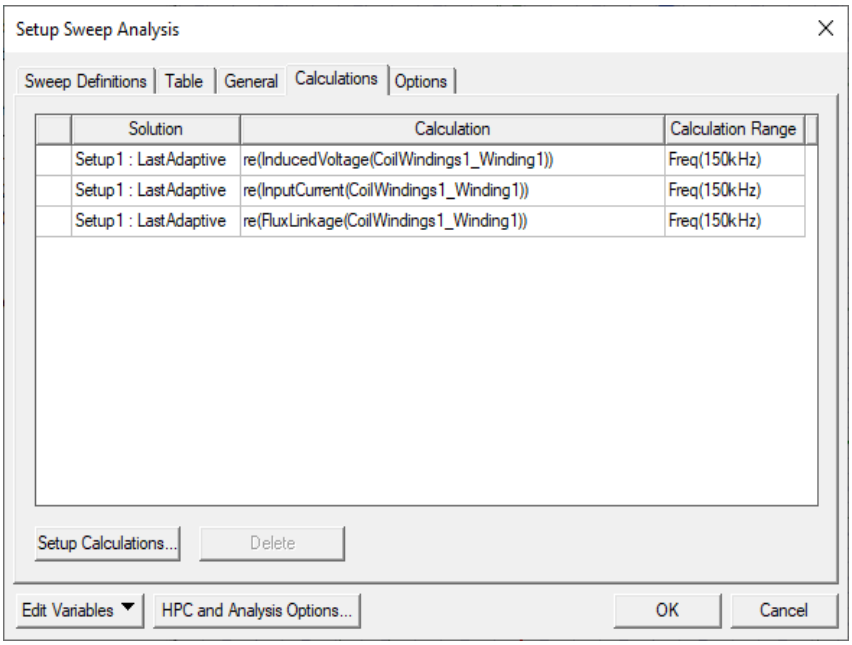
Parameter values can also be edited after the 3D component has been inserted in the design.



- Windings defined in a 3D component are available for use in Matrix setups, Reports, and Optimetrics.







Related Topics

Create 3D Component: Excitations Tab

Assigning a Winding Setup for a 3D Eddy Current Solver

Assigning a Winding Setup for a 3D Transient Solver

Assigning a Winding Setup for a 2D Eddy Current Solver

Assigning a Winding Setup for a 2D Transient Solver

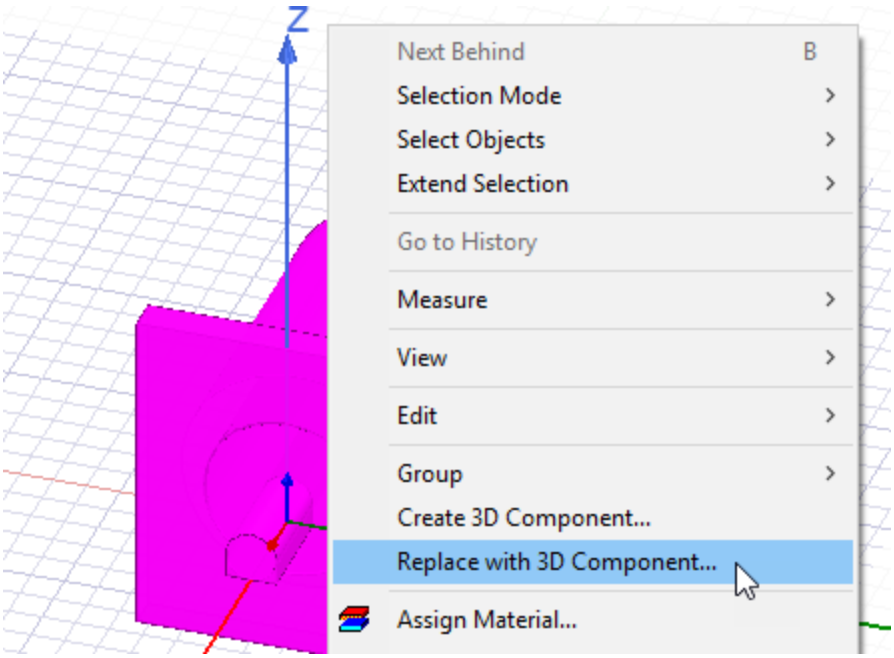
Assigning a Coil Terminal for a 3D Eddy Current Solver

Assigning a Coil Terminal for a 3D Transient Solver

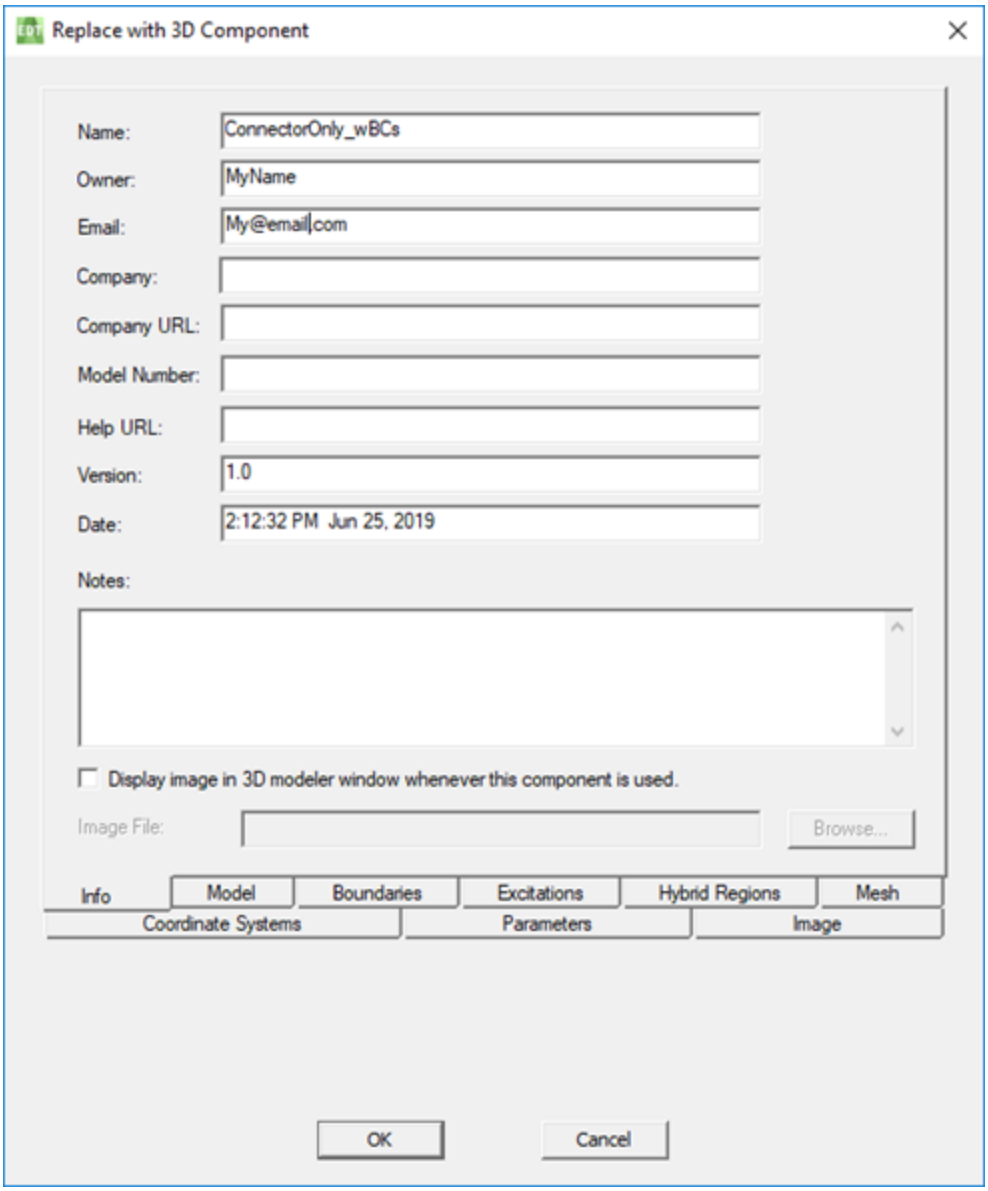
Assigning a Coil for a 2D Eddy Current Solver
Assigning a Coil for a 2D Transient Solver
Assigning a Matrix
Creating Reports
Optimetrics

Replace Selection by 3D Component

To replace a selection in the 3D modeler window with a 3D component version of the selection, select an existing object or model, right-click, and select **Replace with 3D Component**. You can also click **Draw> 3D Component Library> Replace with 3D Component**. You cannot replace components from a design with an array setup. However, you create a 3D component from a unit cell and use that to create a Multi-Unit Cell array. When you replace a selection, object names are retained, port names are retained, component parameters map to existing design variables and report, field plot, and edit source are all preserved.



This opens the **Replace with 3D Component** dialog.



The tabs let you view the component features organized according to geometry, design data, and settings. The **Replace Selection with 3D Component** dialog box does not offer tabs for encryption or licensing, compared to Creating a 3D Component. You can use these tabs to change inclusion of objects, boundaries, ports, coordinate systems, or parameters for the 3D component. When you click **OK**, no *Save* dialog box is shown.

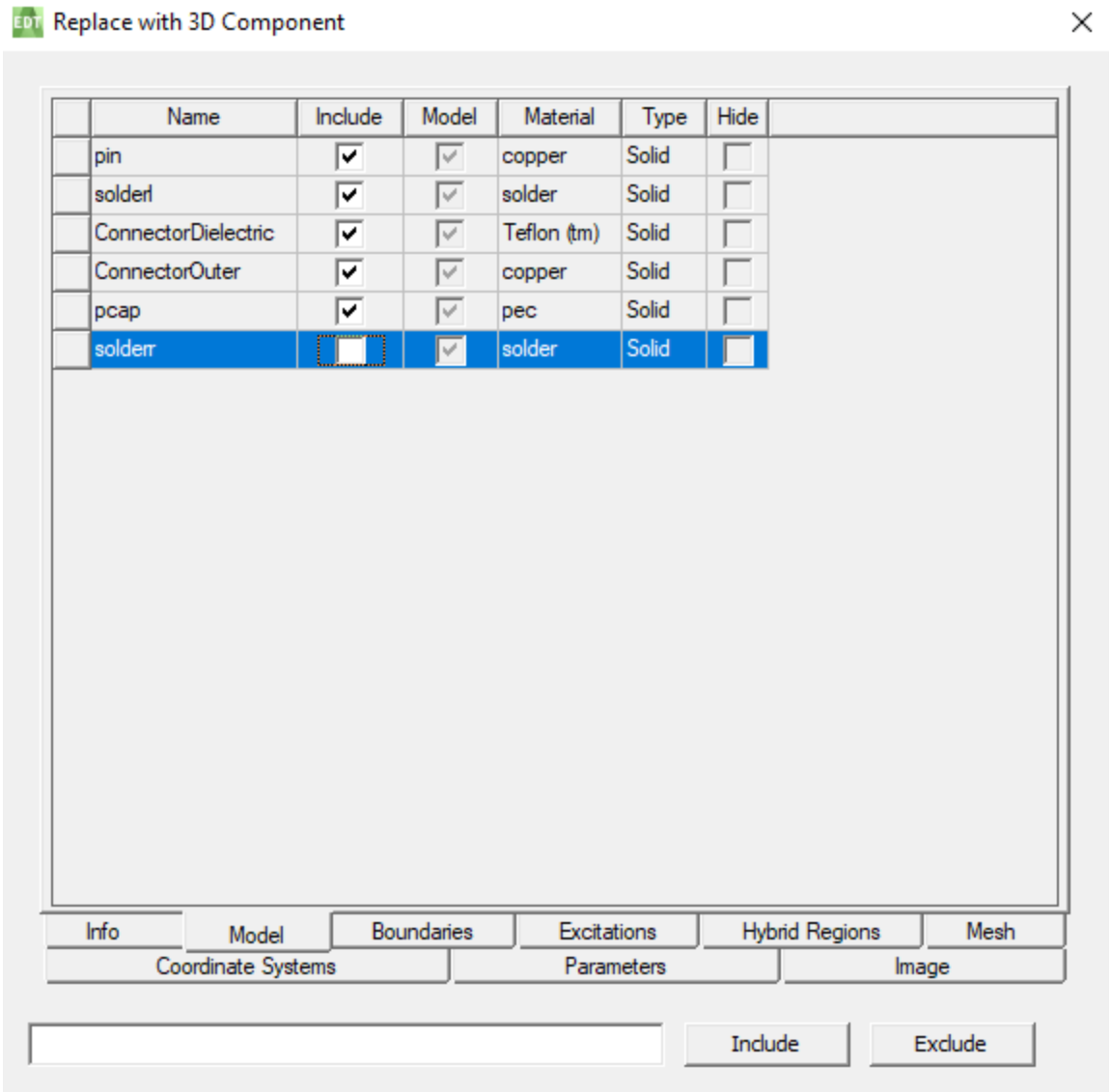
- Info tab
- Model tab
- Boundaries tab
- Excitations tab
- Mesh tab

- Coordinate Systems tab
- Parameters tab
- Image tab

When you click OK, no save dialog is shown. The 3D component is saved to a temporary location. Insertion of instances uses the temporary file, but works just as regular 3D component insertion. The Project tree shows the Component under 3D Components, the boundaries are no longer listed under the Boundaries icon. Excitations, Reports and Analysis Setups remain the same. History Tree no longer shows detailed model edits. You can **Undo** and **Redo**.

Include or Exclude Feature from Component

You can view the various tabs and select which features to include or exclude while replacing the selection with a component. You can use the check boxes in the **Include** column for this purpose. You can also type a feature *Name*in the text box and use the **Include** or **Exclude** button. The text field supports wildcard characters (* and ?), so you can toggle the inclusion state for multiple objects with similar names in a single operation.



An object's **Include** option under the *Model* tab impacts the inclusion state in other tabs. For example, if you exclude an object from the component creation, the design data, parameters, and coordinate systems corresponding to that object are also excluded.

Sorting Columns Using the Headers

The features listed for each tab include columns for the Name, check boxes for whether to include, and properties. You can click on the column headers to sort by Name, Model, or other feature listed for each tab. Click the column header to selects the column to sort. Click again to invert the column.

You can sort lists by using all columns except Include.

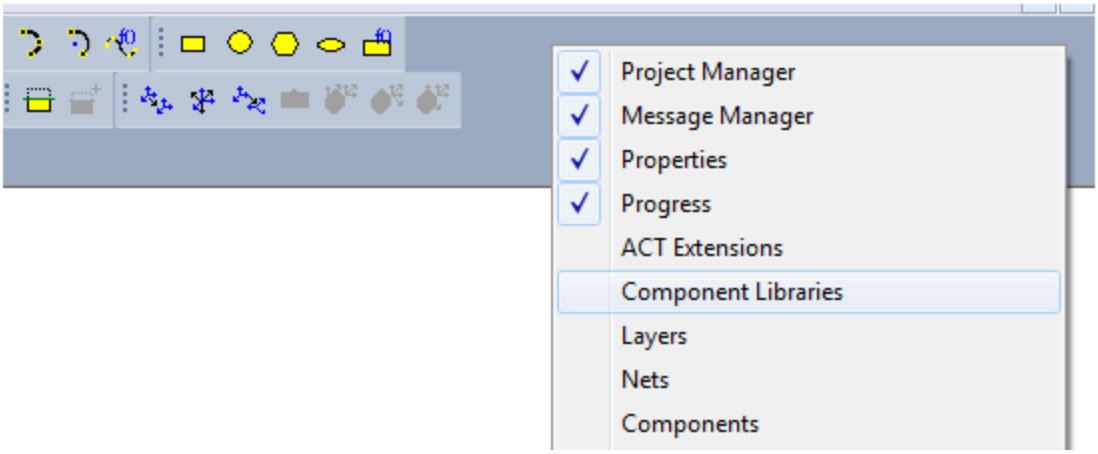
Design Data Notes:

- DC thickness on selected objects is always included in components. It is not listed in the **Boundaries** tab.
- The **Create 3D Component** dialog does not list design data without an assignment.
- The parent of any included boundary/excitation is included, as long as the parent does not require assignment.
- Design settings like material overrides are included.

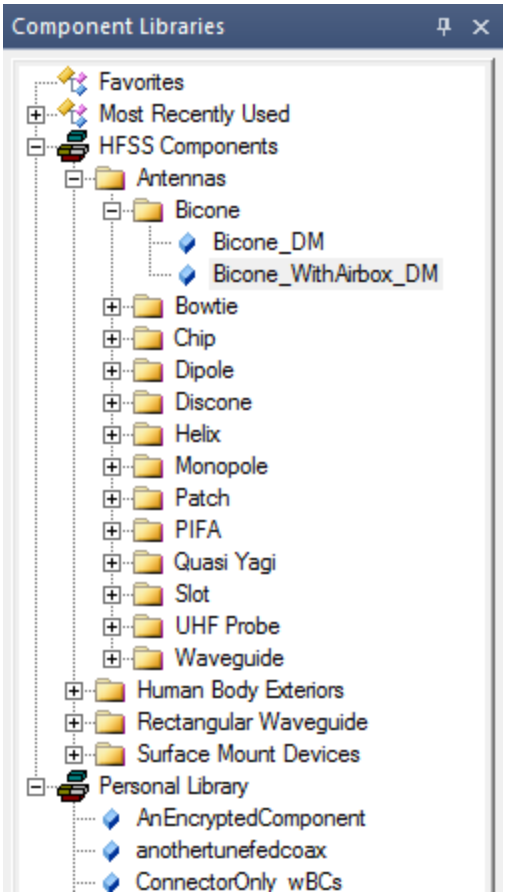
Once you have made the Include and Exclude settings, and have specified the Image and Info you click **OK** to Replace with 3D Component. When you click OK, the component is validated for everything that is included in the component. For example, you cannot include boundaries if the object on which the boundaries have been created is not included.

Using the Component Libraries Window

Once you save one or more components to a library, and create target coordinate systems in the design where you intend to place the component, you can right-click the 3D component icon in the Project tree or use the **Draw** > **3D Component Library** commands to browse your folders or libraries, or use the **View** > **Component Libraries** to display a **Component Libraries** window to navigate installed libraries. Visibility of the **Component Libraries** window is remembered for each design type so that if context switches between different design type, the window is shown or hidden based on setting for new design type.

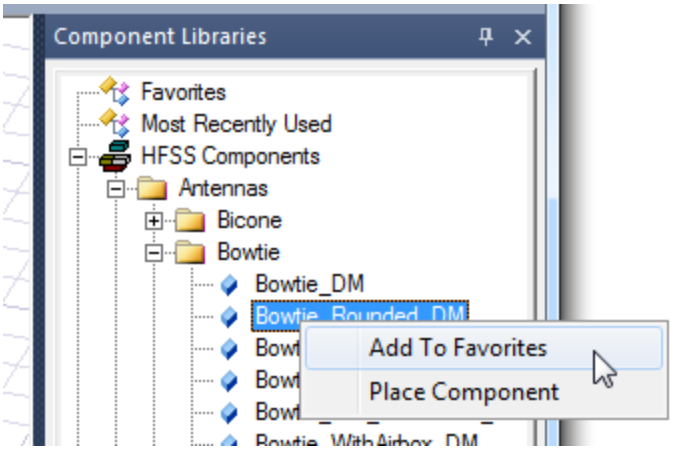


From the **Component Libraries** window, you can navigate the directory of installed components, as well as any in your Personal Library. The **Component Libraries** window lists all the 3D components available in syslib, UserLib and PersonalLib. It adds a folder corresponding to a library only if there are any components for active design under that library. There are also additional optional folders for managing Favorites and Most Recently Used components. **Component Libraries** window contents, including Most Recently Used and Favorites correspond to active design. When context is switched between different designs, Component Libraries window will be refreshed so that it is relevant to active design.



You can control whether to show the Favorites and Most Recently Used folders from the **Tools** > **Options** command dialog box, on the General > Component Libraries page.

Right-click a component to view the **Add to Favorites** command to include frequently used components to the Favorites folder. Once a model is included in the Favorites folder, you can also right clock to **Remove from Favorites**.



When you have added a component to a design, it also appears in the Most Recently Used folder. You can also right-click model there to view the **Add to Favorites** and **Place Component** commands.

You can select any component from the library, and drag and drop to insert it in the design. If you insert a component by the drag and drop method, the component is immediately inserted, skipping the **Insert Component** dialog box. The component is inserted with default component parameter values and is placed in current active coordinate system.

You can also select **Place Component** from the right-click menu or double-click any model in the component libraries to view the **Insert 3D Component Instance** dialog opened to the **Parameters** tab. Using the **Place Component** menu command allows you to review component parameter values, change target coordinate system, and so forth, using the **Insert Component** dialog box. You can also view the **Image** and **Info** tabs.

Component file organization on the disk

3D Component files are organized under three libraries, namely *syslib, UserLib,* and *PersonalLib*. 3D component files must be in a **3DComponents** folder under the appropriate library location. It is recommended that files are further organized by design type under the *3DComponents* folder. For example, component files for Maxwell should be organized under 3DComponents\Maxwell, as done for syslib. The 3D Component Libraries window and also Draw\3D Component Library\ menu look for components under a folder corresponding to the active design type, and it ignores component files under folders corresponding to another design type. For example, Maxwell will ignore component files under the HFSS subfolder. Any component files that are directly under *3DComponents* will be visible in all design types.

Related Topics:

Show, Show Only, Hide and Fit features, helpful in working with 3D components are described here:

Fit All Objects in a View Window.

Fit Selected objects in a View Window.

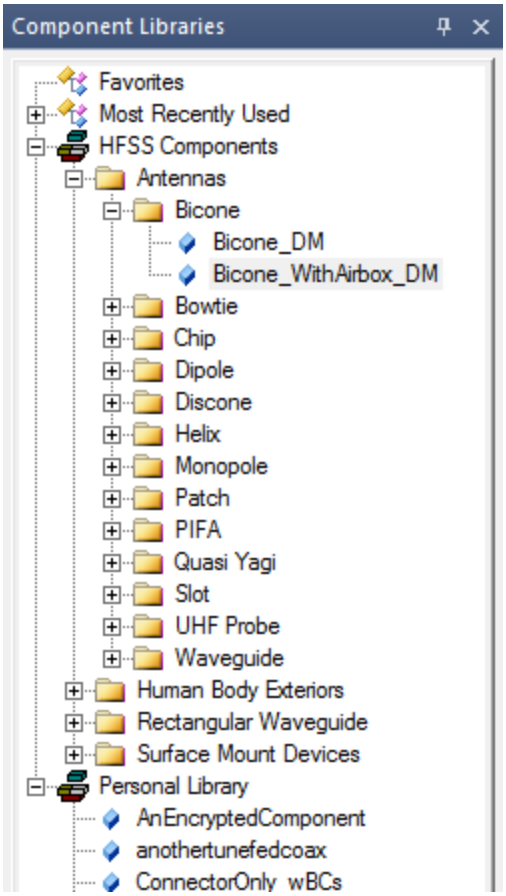
Showing Only Selected Objects in All or Active Views

Hiding Objects from View

Inserting a 3D Component in a Design

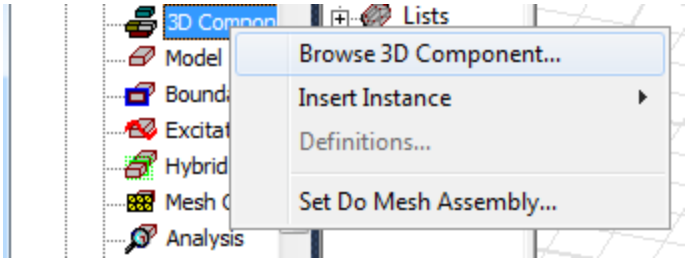
Once you save one or more components to a library, and create target coordinate systems in the design where you intend to place the component, you can either right click on the 3D component icon in the Project tree or use, or the **Draw>3D Component Library** command to browse your folders or libraries, or use the **View>Component Libraries** to display a Component Libraries window to navigate installed libraries.

From the Component Libraries window, you can navigate the directory of installed components, as well as any in your Personal Library. The Hierarchical tree display includes Favorites and Most Recently Used branches.

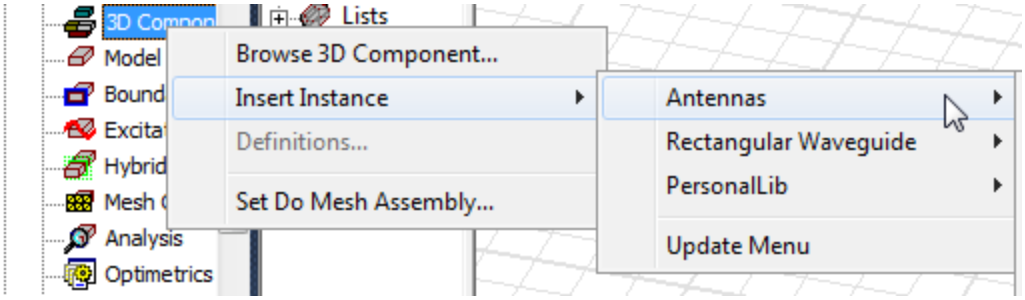


You can select any component from the library, and drag and drop to insert it in the design. Double-click on any model in the component libraries to view the **Insert 3D Component** Instance dialog opened to the **Parameters** tab. You can also view the **Image** and **Info** tabs, and select the target coordinate system.

Selecting **Browse 3D Components** lets you navigate directories via a browser window.

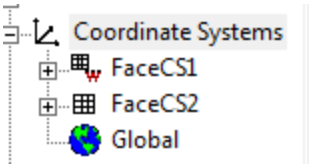


Selecting **Insert Instance** displays further menu selections for pre-defined components in other libraries such as the Antennas library and the Rectangular Waveguide library. If you have added components to PersonalLib or UserLib, these also appear after you refresh the menu.

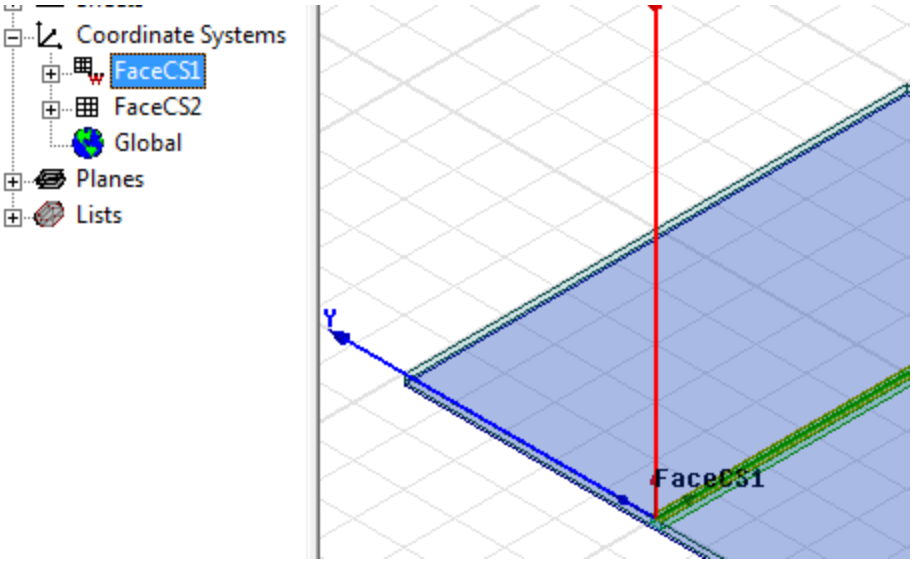


You can also click **Draw>3D Component Library** to access the **Browse** and **Insert Instance** menus to select a component to insert into a design.

Before inserting a component in a design, it can be helpful to create a target coordinate system to provide a location for the component. In the following 3D project example, the design includes two additional coordinate systems:

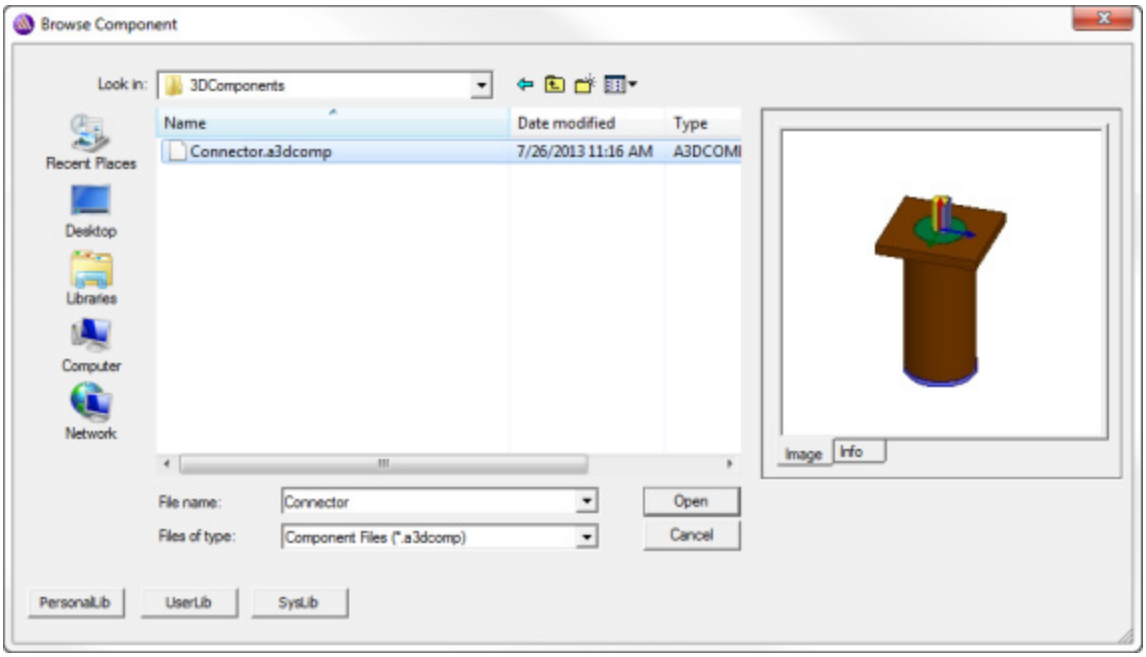


Selection of a coordinate system in the History tree displays a potential location for inserting a component.



To Insert a 3D Component from the Menus

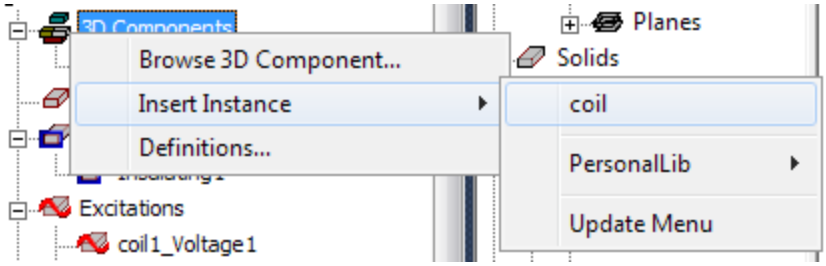
1. Right-click on the **3D Component** icon in the **Project tree** to open shortcut menu. You can select **Browse** to use a browser window navigate the file system to the component.



Selecting the *.a3dcomp file displays the component image and file name.

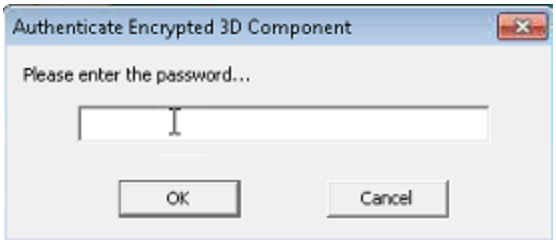
Note	3D components created in Maxwell 2D cannot be used in Maxwell 3D projects, and vice-versa. A message displays if you attempt to insert a component that is incompatible with the active design.

If you have previously inserted a component into a Project, the shortcut menu for the 3D Component displays that component so that you can easily insert another instance of the same component.



2. If you have used Browse, select a component and click the **Open** button. If you use the Insert Instance menu, select the component name.

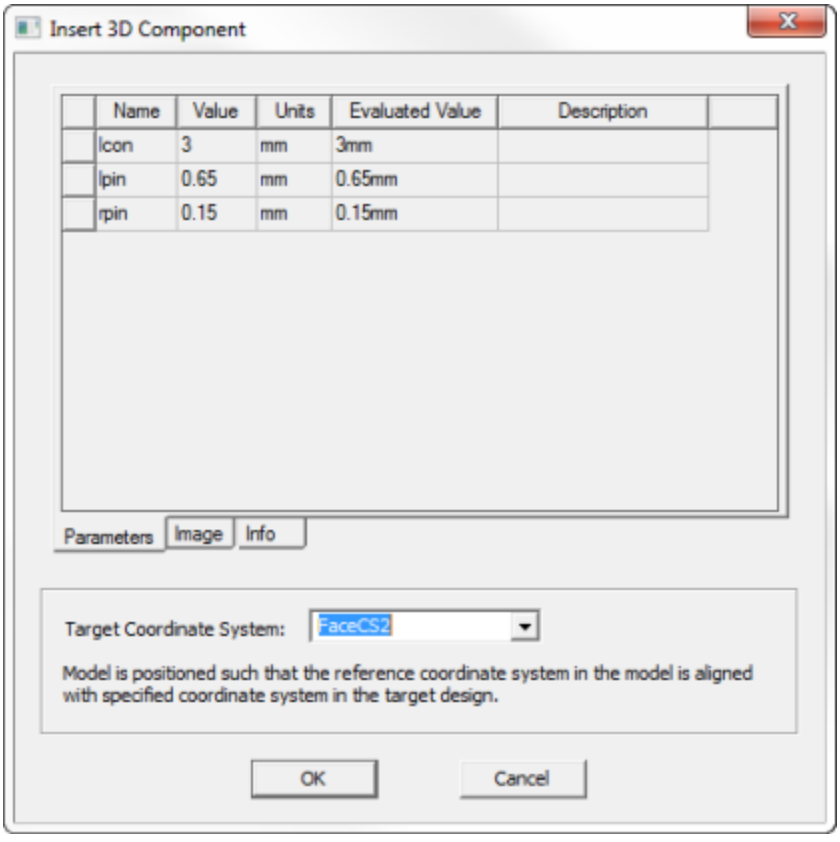
   If the component has been encrypted, you may be prompted to enter a password. If you insert another instance of the same component, you do not need to enter the password again for that component again.



You have three tries to enter the correct password. Upon successfully entering the password, the Insert 3D Component dialog opens with the Encryption tab displayed.

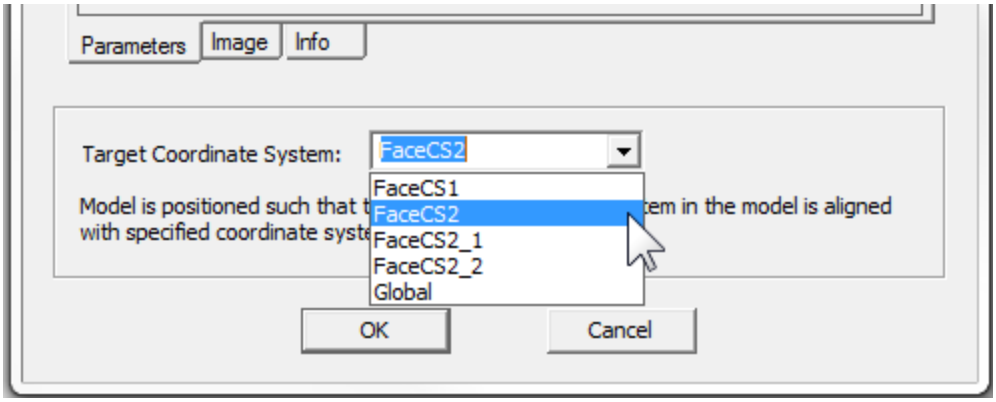
Note: There are some additional considerations with respect to projects using encrypted components. When a project contains encrypted components, the project file and other related files are also encrypted. Encrypted component contents may also be hidden from view in the design where the component is used. For more details, see Create 3D Component Encryption Tab.

You see the **Insert 3D Component** dialog opened on the Parameters tab.

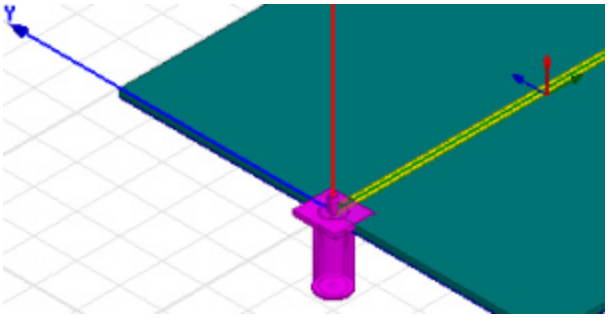


Tabs let you view the Parameters, Image, and Info. You can edit parameter values, and assign variables or expressions for parameters

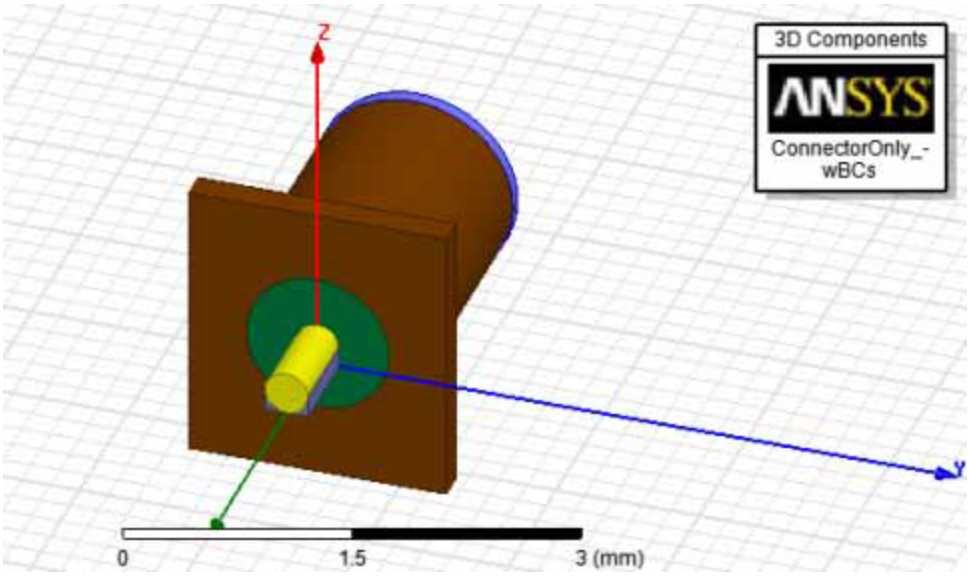
3. Use the menu to select the Target Coordinate System if any have been defined in addition to the Global coordinate system. The target coordinate system that you select is highlighted in the modeler window.



4. When you click **OK** the component is placed at the coordinate system you selected.



5. If the component has a logo defined, the image is always displayed in the upper right of the modeler window.



Related Topics

Show, Show Only, Hide and Fit features, helpful in working with 3D components are described here:

Fit All Objects in a View Window.

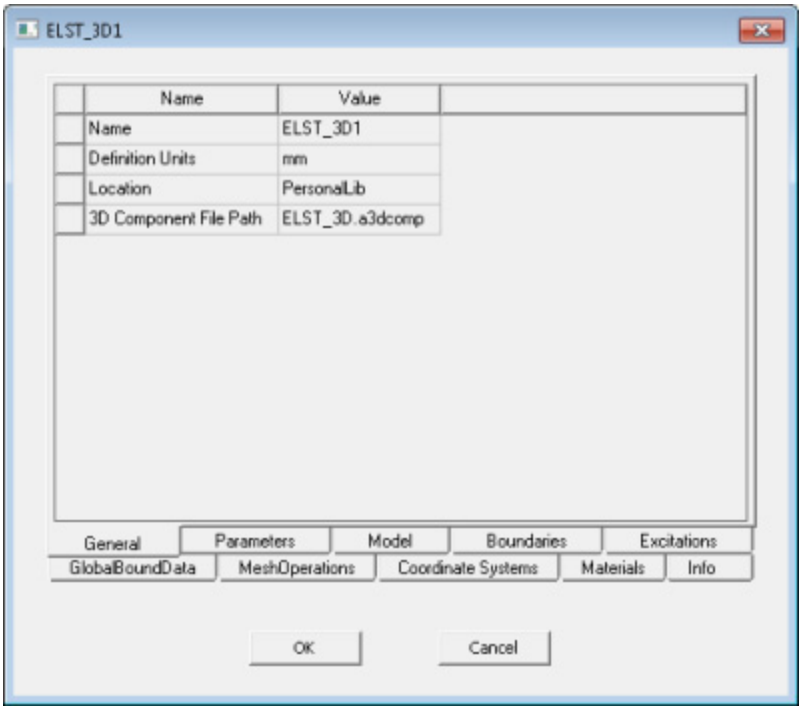
Fit Selected objects in a View Window.

Showing Only Selected Objects in All or Active Views

Hiding Objects from View

Viewing Properties of 3D Components

The Properties displayed for a selected 3D component depends on how you select that component. For a complete display of Properties, right click on the selected component (Project tree, History tree, or Modeler window) and click **Properties** on the shortcut menu, or click **Edit>Properties**.This displays a Properties window with tabs for all component properties.

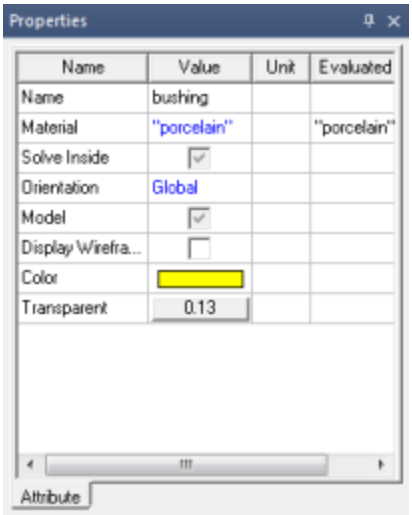


Selecting a 3D component in the Project tree displays docked properties with tabs for General, Materials and Component data.

If you select a component in the **Project** window, then right-click and select **Edit>Properties**, you can view the **Properties** window. If you choose the **Materials** tab, you can then select any listed material and the parts of the component made of that material are highlighted in the **Modeler** window.

Selecting a 3D component in the History tree displays docked properties with tabs for General and Parameters.

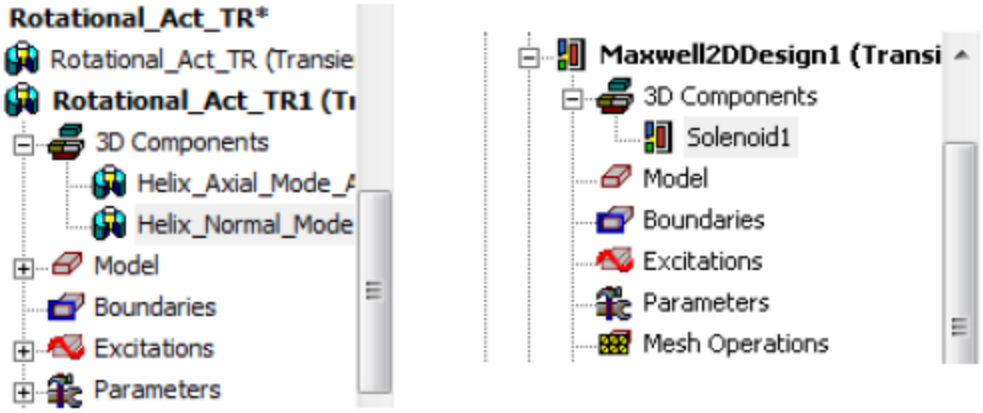
Selecting a 3D component in the Modeler window displays docked properties with a tab for Attributes.



Component materials do not appear in the Project Tree Definitions list. If you select the icon for an inserted component, the Properties window contains a Materials tab that shows the Materials defined for the selected component.

3D Components in the Project Manager Window

The **Project Manager** window organizes any components that you add to a design under the 3D Components icon in the **Project** tree.

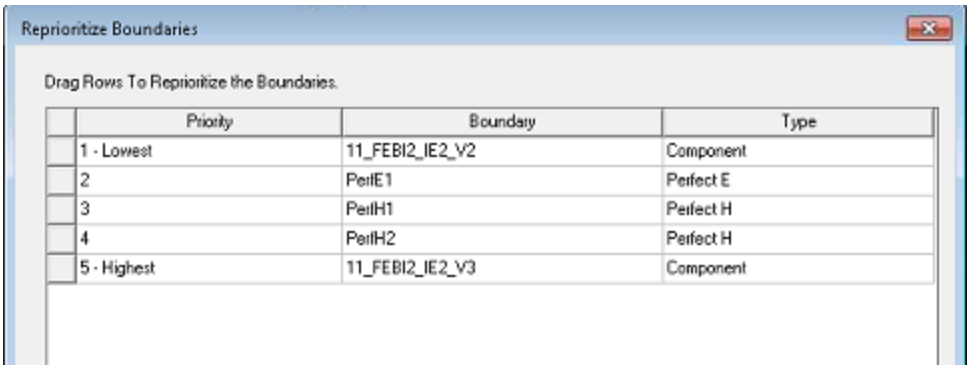


Selecting these inserted components populates the Properties window which has tabs that display the instance attributes.

Excitations included in the 3D Components appear in the Project tree. This gives you the ability to edit Post Processing settings.

Boundaries defined inside the component do not appear in the project tree. However, boundary properties can be when you select the component in the Project tree and view the Component properties under the Component Data tab.

Also, you can use **Maxwell>Boundaries>Reprioritize** to set the priorities of all boundaries relative to non-component boundaries.



Related Topics:

Fit All Objects in a View Window.

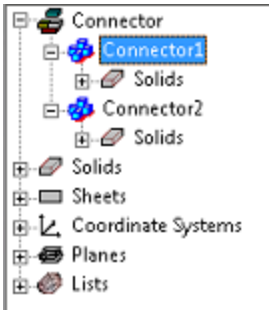
Fit Selected objects in a View Window.

Showing Only Selected Objects in All or Active Views

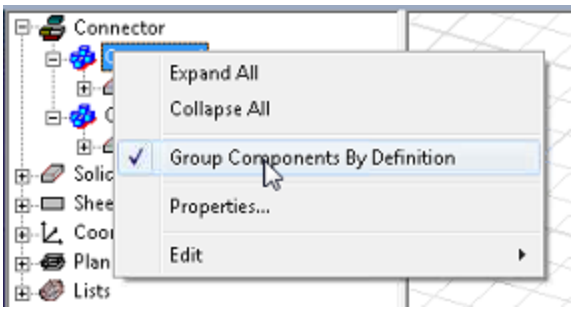
Hiding Objects from View

3D Components in the History Tree

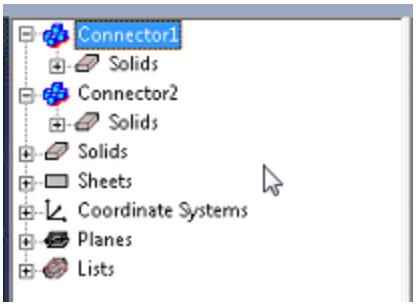
You can group all instances of a component by component definition in the History tree.



To disable this grouping, right click on an instance, and uncheck Group Components by Definition.

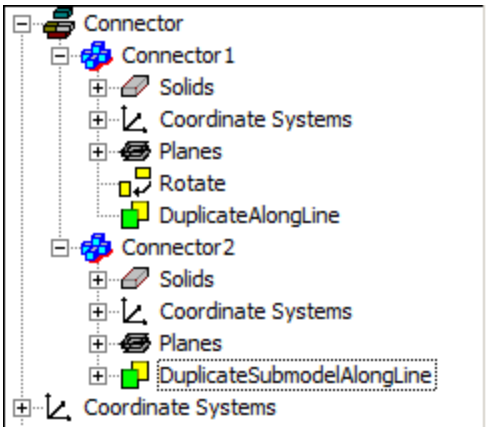


This causes the display to show numbered instances listed without the icon for the primary component.



Component Operations in the History Tree

- Component operations appear under that component folder toward the end.



- Component operations act on parts as well as CS.
- Boundaries and mesh operations are duplicated with duplicate and copy/paste operations, provided that you check the **Duplicate boundaries/mesh operations with geometry** option to do so (see Tools>Options/<design type options>).

Related Topics

Fit All Objects in a View Window.

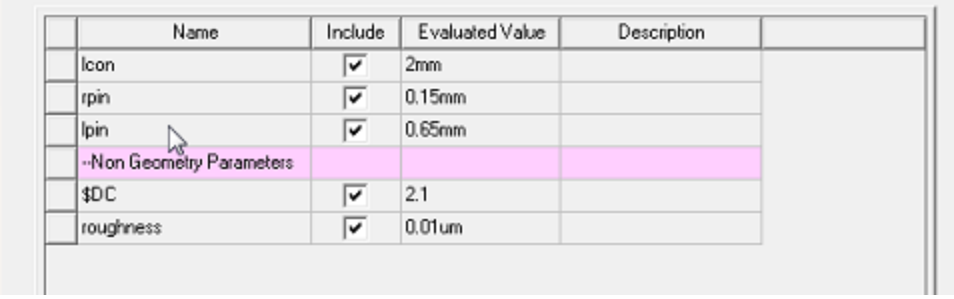
Fit Selected objects in a View Window.

Showing Only Selected Objects in All or Active Views

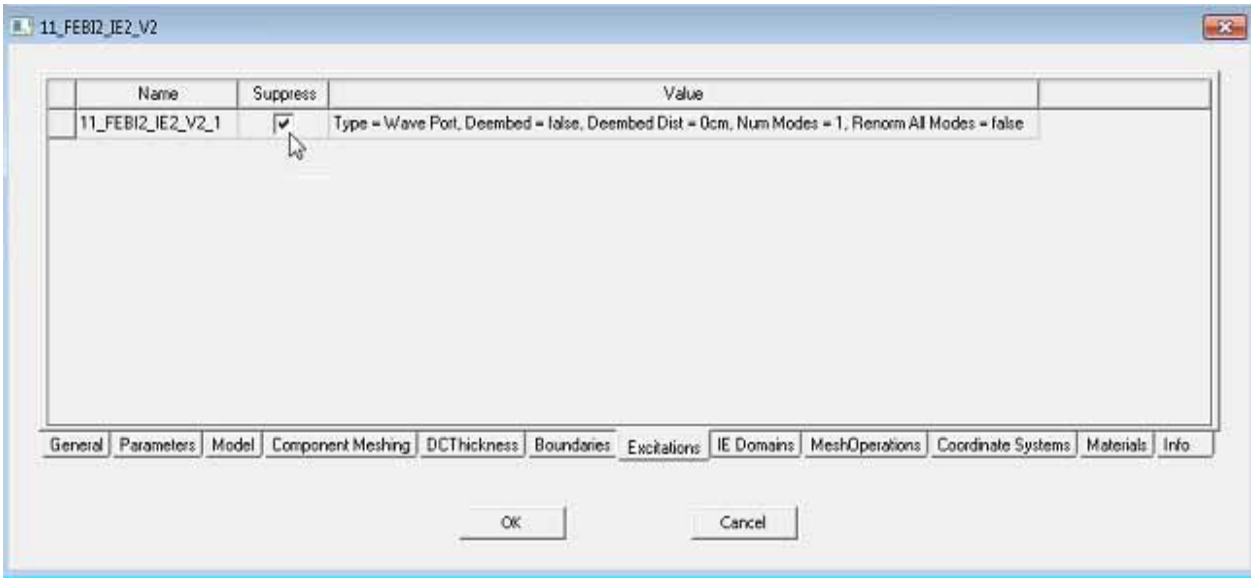
Hiding Objects from View

Editing 3D Component Properties

- Edit component parameters, if defined for that component. Editing the parameters of one component instance affects only that instance. If a component includes Non-Geometry Parameters, these are grouped separately on the Parameters tab.

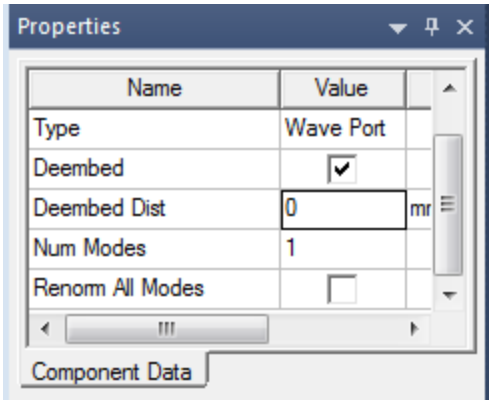


- Edit excitations by suppressing their use on the Excitations tab in the Properties window.

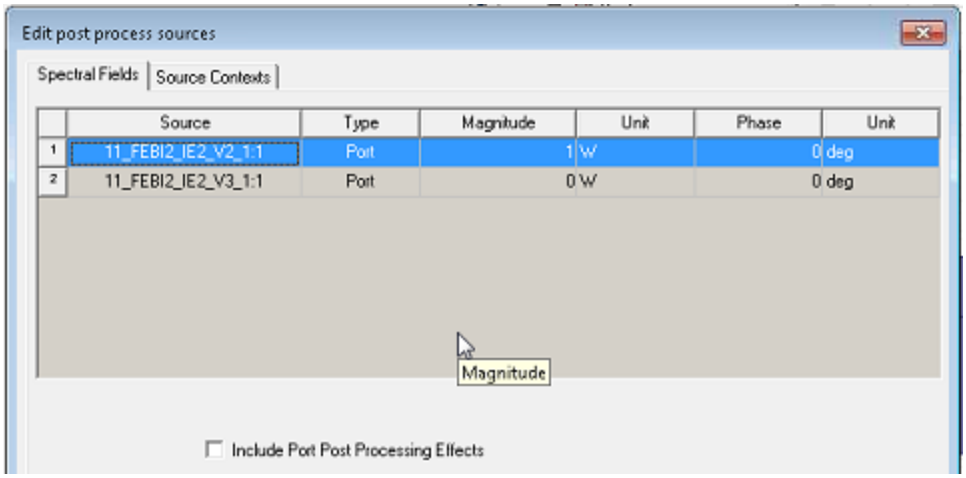


Checking Suppress in the Properties window removes the port on the selected instance from the Excitations list on the Project tree. Unchecking Suppress activates that port for the model and causes the port to appear in the Excitations list.

- Edit excitation Post Processing such as deembed distance, renormalization.



- Edit Sources for Post Processing



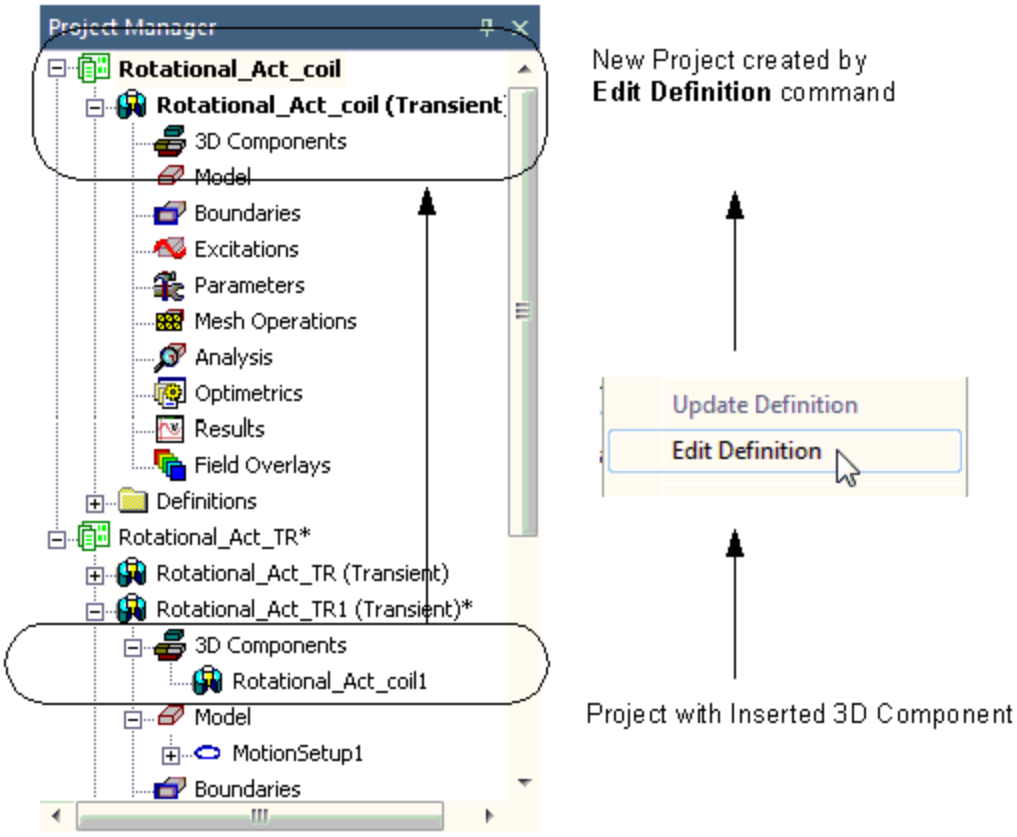
- Edit component attributes like component name, part names, material assignment, and model/non-model flags.
- Edit boundaries, excitation and mesh operations for the component on the corresponding tabs in the Properties window.
- Add additional boundaries/excitations on parts coming from component

  However, operations on individual parts of component are not allowed.
- Copy/Paste component
- Delete component
- Arrange component: Move, Rotate, and Mirror
- Duplicate component: Along Line, Around Axis, and Mirror
- Edit Attributes for Display Wire frame, Color, and Transparency.
- Use **Edit>Selection Mode>Submodels** to quickly select component

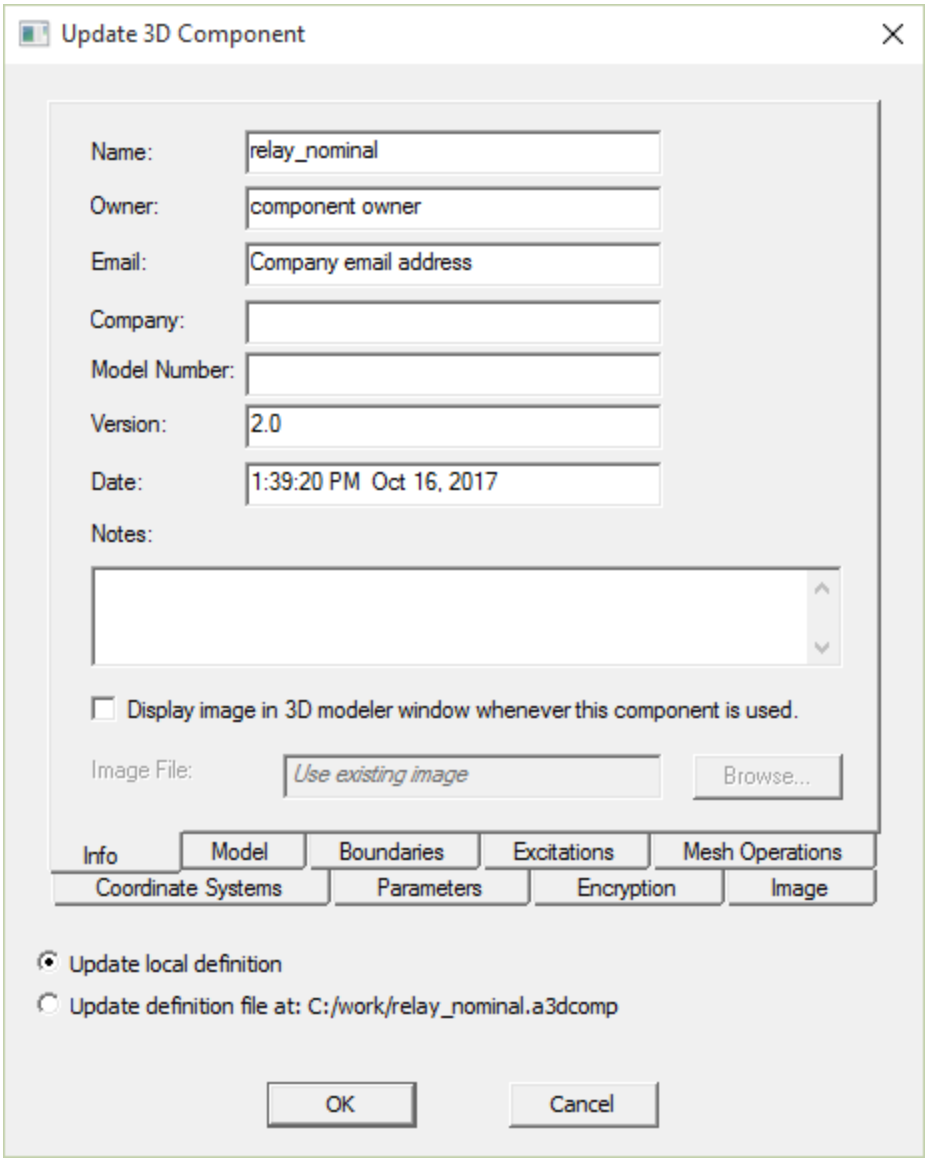
Editing a 3D Component Definition

The Edit Component definition feature allows you to create new version of component definition just by using previous 3D component file. This should also allow you to quickly create new version without having to go through all the creation steps. For 3D Components that are not encrypted you can edit the definition. Software will automatically create new project in the existing session. The Project will have one design of same type as original design from which edit definition was initiated. All 3D component data from the file will be read in ‘exploded’ form in that design. Component materials will be added at the project level, variables will be added at the project and design level, geometry will create detailed history for all objects in the component and design data will be added to respective folder in the project tree in the detailed form.

The **Edit Definition** command creates a new project named for the component. You can see the project in the **Project Manager** window.



You can then edit this new project. When you save the edited component, you see an **Update 3D Component** dialog including radio buttons for whether you want to save the edited component as a local instance, or to the original library.

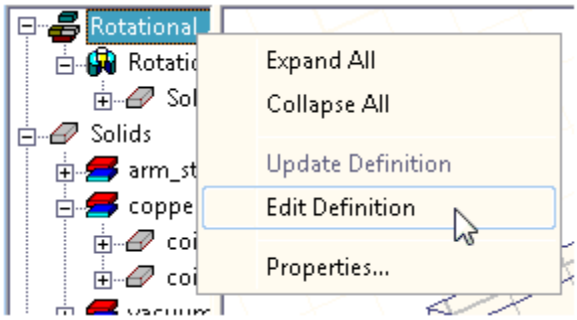


If you select **Update local definition**, any changes you make apply only to the local file. If you choose **Update definition file at: <*path*>** to save the component to the definition file in your library, you can Update 3D Component Definition for other component instances inserted in your designs.

To edit a 3D Component definition.

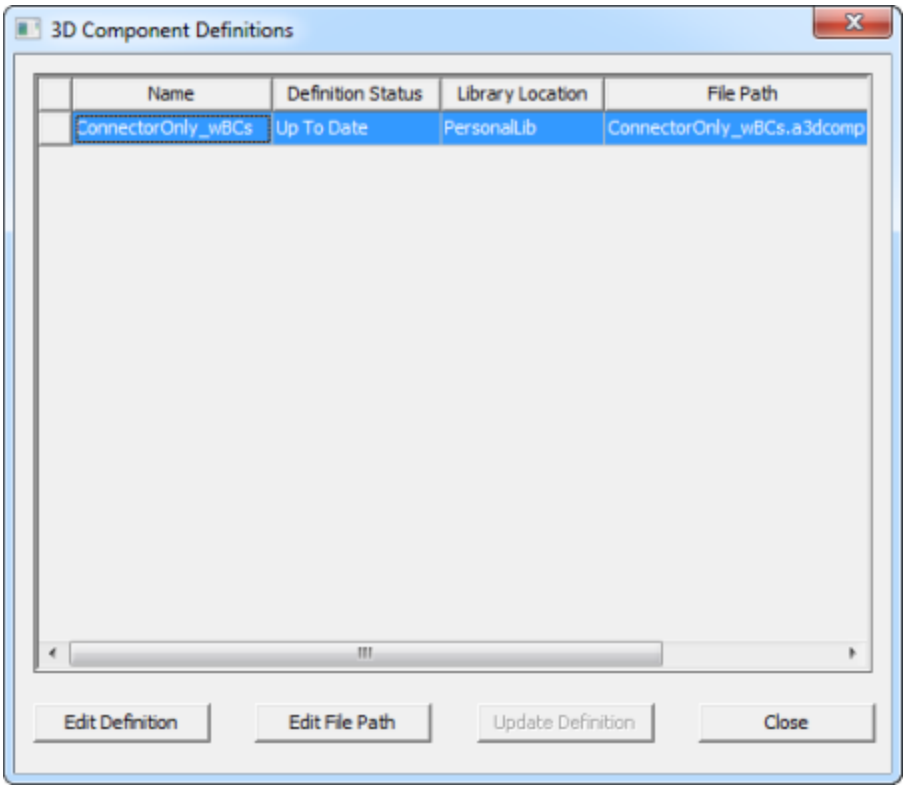
1. You can begin editing a 3D component definition from either the History tree or the Project tree.
   - To begin from the **History tree**, select the component,, and right-click to display the shortcut menu and click **Edit Definition**.

   For unencrypted components, this creates a new project in the **Project window** named for the component. For encrypted components, you must first supply the edit password.



- To begin from the **Project window**, select the 3D Components icon and right click to display the short cut menu. Then select **Definitions**.

This displays the **3D Component Definitions** window in which you can select the component and then click **Edit Definition**.

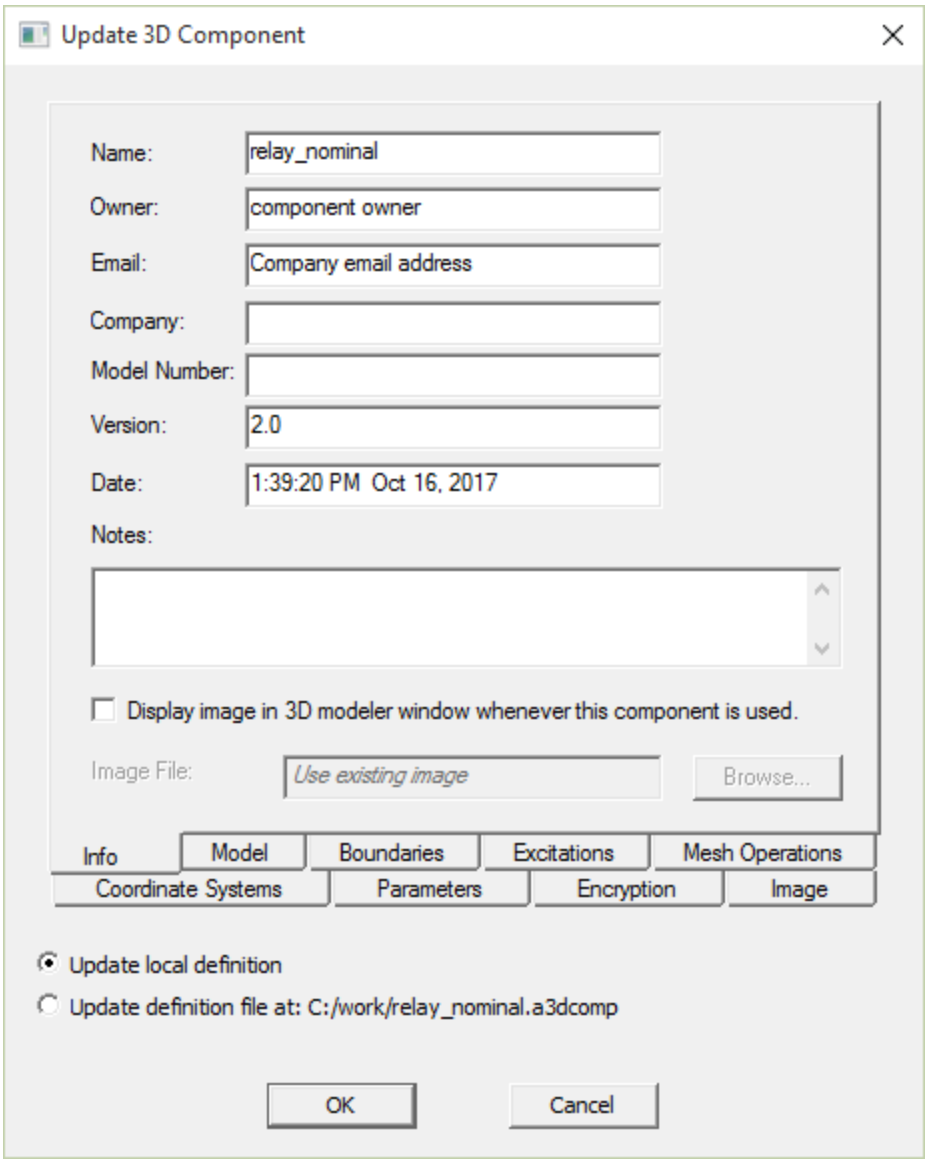


2. For 3D Components that are not encrypted, you can edit the definition. Software will automatically create a new project in the existing session. The Project will have one design of same type as the original design from which edit definition was initiated. You can now edit the design with component data as required without any restrictions. All the component details are available in their respective folders. Complete geometry history is available in

history tree. Material properties are added to the project definition folder. Boundaries, excitations, etc., go to their respective folders in the project tree and component parameters become project or design variables.

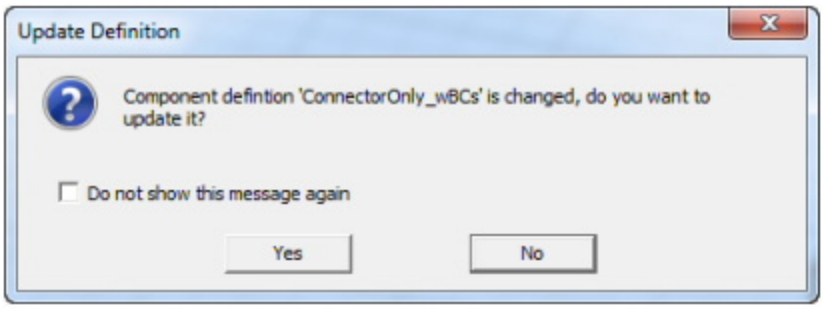
3. You can use **File>Save** to update the original component definition file. The **Update 3D Component** dialog will come after **File>Save**.

   The **Update 3D Component** dialog is populated with all entities - objects, boundaries, excitation etc. in the “component-edit” design. In addition, previous settings like reference coordinate system, component name, owner etc. from the **Info** tab, as well as encryption settings, are preserved. You can also select whether to **Update local definition** (for local editing) or **Update definition file at: <*path*>** to update the component library, which will affect other instances when updated.

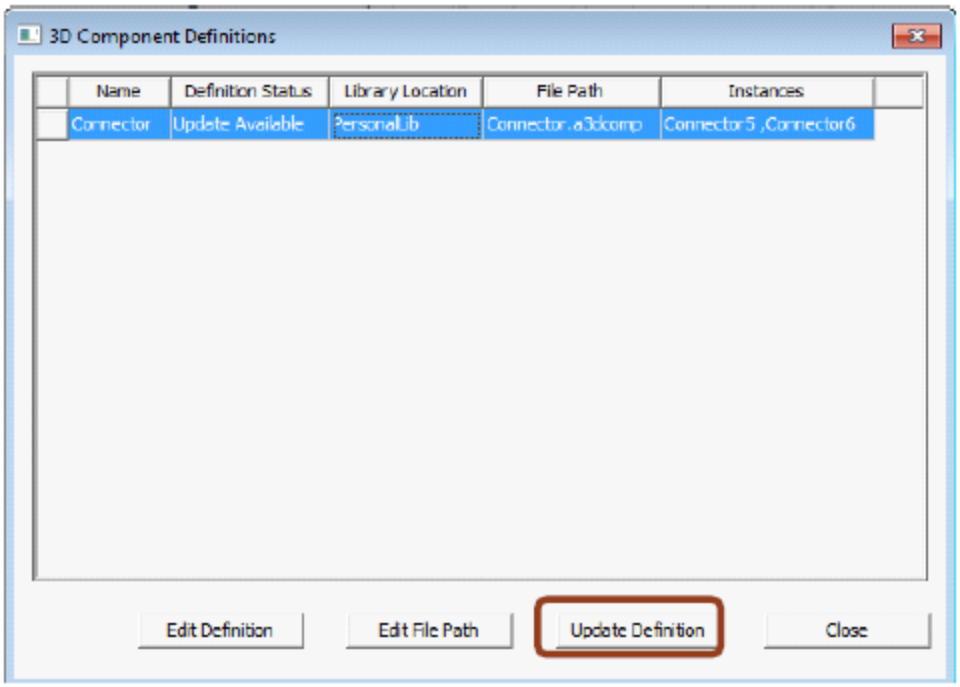


If you use **File/Save As** to save to a new component definition file, you also have the option to **Save As** Ansys Electronics Desktop file.

4. Once you have saved the updated component version, you can go back to the original design and update all the instances using that component. Note that update definition is not automatic even though component definition edit was triggered from that design. Instead when an assembly design becomes active, it prompts you to update the 3D component definition. You can then select to update the definition.



You can also open the 3D definitions window to select a component, and view the Definition Status column. If an update is available, you can select the component and click **Update Definition**.

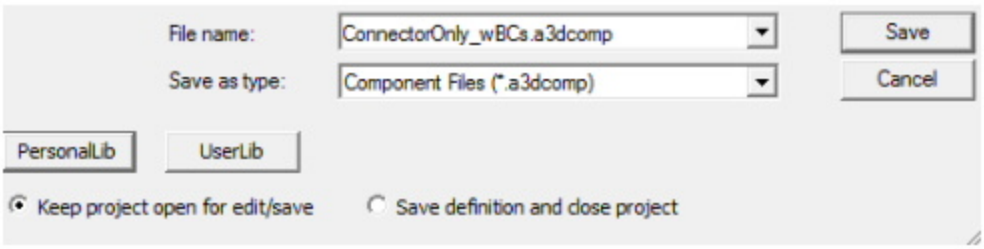


5. See Update 3D Component Definition

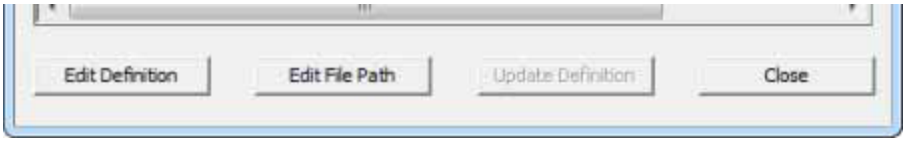
6. After editing is done, you also select the model and then use the **Create 3D Component** command.

   See Creating a 3D Component from an Existing Model.

   The component data is loaded from component definition file. The Info tab version automatically increments. Before saving, you have the option to **Keep project open for edit/save**, or **Save definition and close project**.

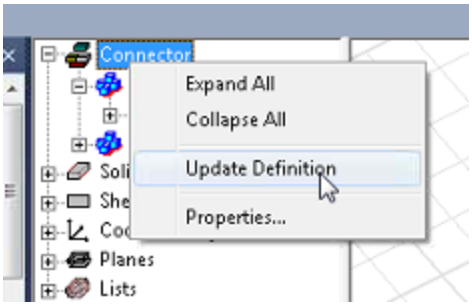


7. You can select **Edit File Path** to update the file path for component definition. This opens a browser window that lets you navigate your file system and select a file path to be used for the new component definition.

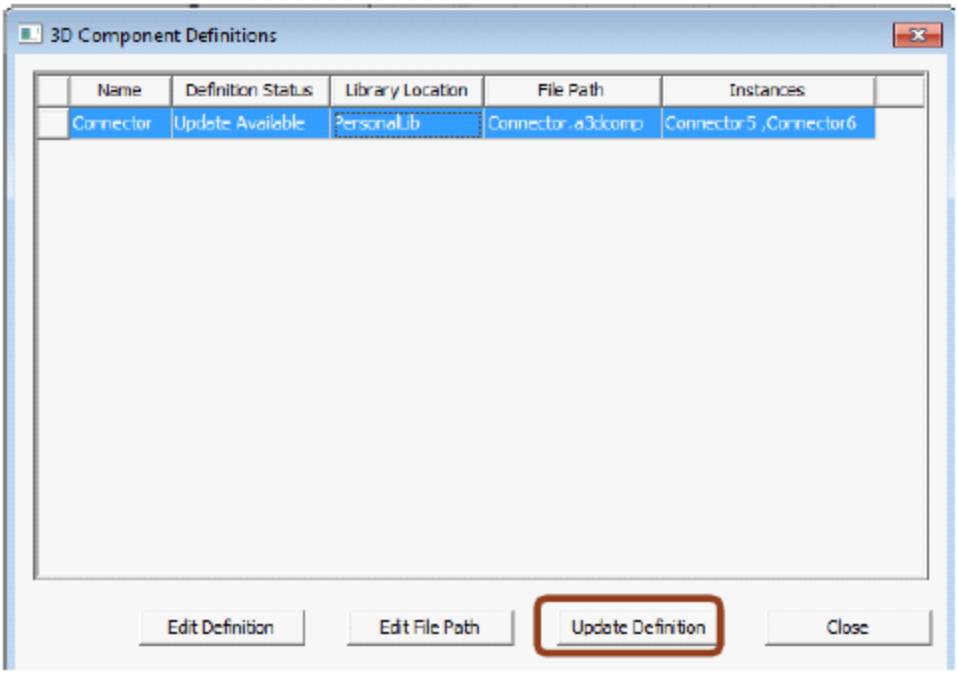


Updating 3D Component Definitions

If you create or obtain from a vendor a new version of a component, and then open a project that contained an earlier version of that component, you can right-click on the component in the History tree and select **Update Definition** to replace the older version of the component with the newer one This updates all the instances of the component using that definition. Update Definition is grayed out if a newer definition is unavailable.



You can also accomplish this by right-clicking on the 3D Component icon in the Project Tree and selecting Definitions from the shortcut menu. This displays a dialog listing available component definitions. The Definition status column lets you know if an undated definition is available.

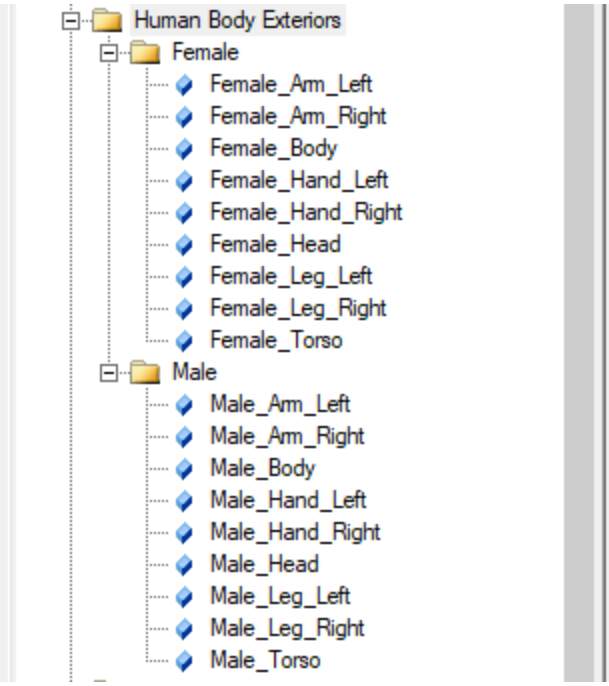


All instances and component operations in the design are updated

- Parameter values are preserved
- New parts are generated in all instances
- Deleted parts are deleted from all instances
- Edited parts are changed in all instances
- Design data, materials, coordinate systems can also be updated
- Undo is supported, command is scriptable
- Definitions dialog provides complete list of definitions in the design
- Definitions can be updated from the dialog

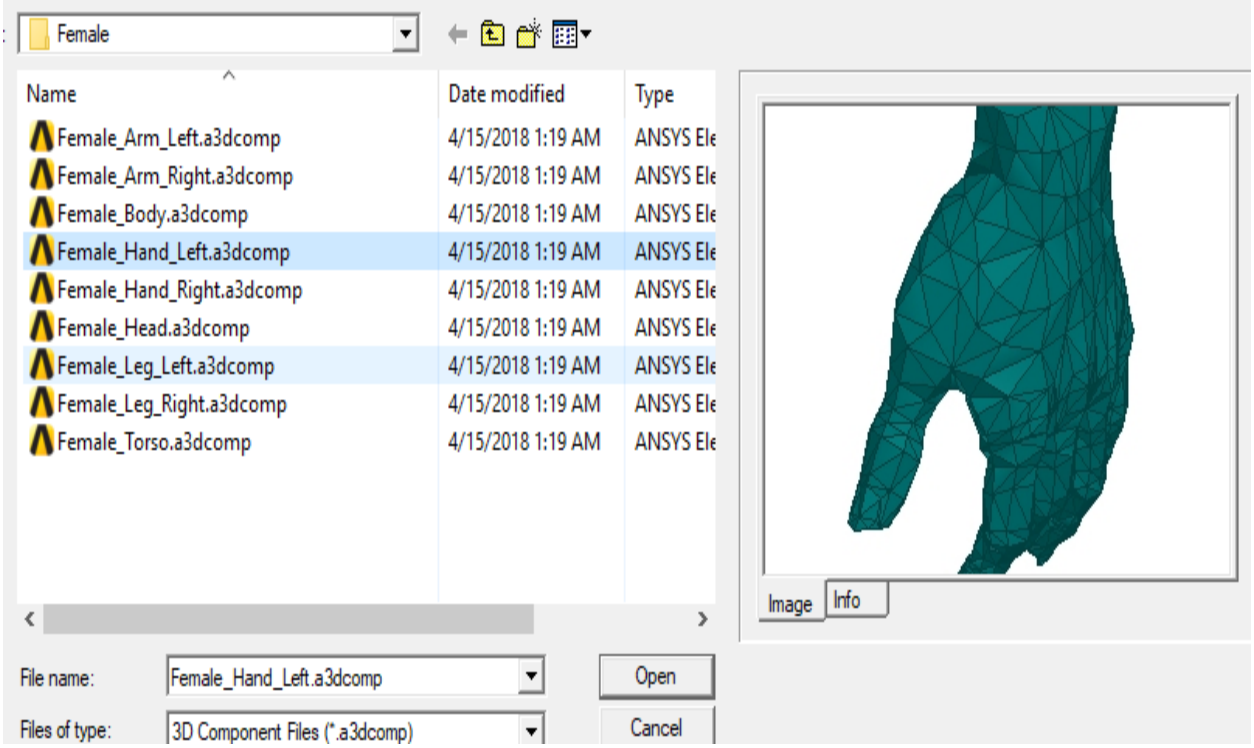
Human Body Exteriors

There are two ways to view the Human Body Exteriors in the 3D Component Libraries. If you have enabled **View** > **Component Libraries**, you can navigate to the different models in the tree. From the **Component Libraries** tree, you can drag and drop models to the **Modeler** window.

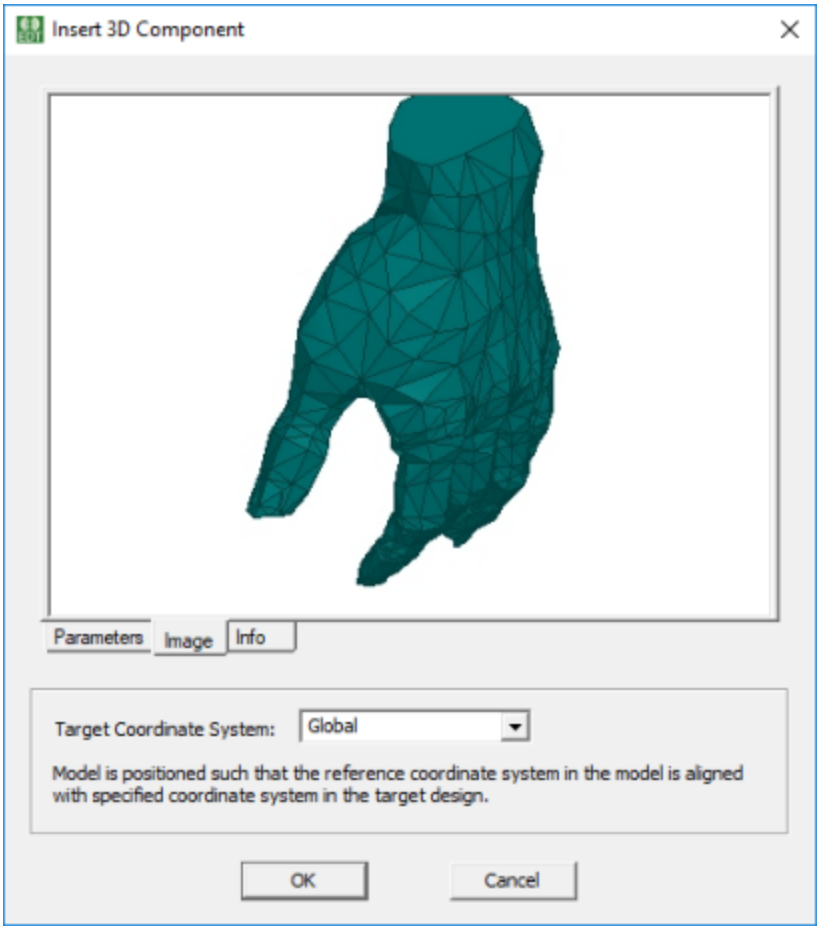


You can also access the **Human Body Exteriors** library using **Draw** > **3D Component Library** > **Browse...** to open the **Browse 3D Component** dialog. you then click **SysLib** to display the libraries included in your installation.

You can select from the Human Body Exterior models, and navigate the folders to select from available components. Once you have selected a Component file, you can view the Image and Info for that component.



You can click **Open** to display the **Insert 3D Component** dialog box. You can view the Image and Info tabs for that component. Human Body Exteriors do not have parameters.



You can also open the **Insert 3D Component** dialog by double-clicking on a model in the **Component Libraries** window. You can view the Image, and Info tabs for that component. See Inserting a Component into a Design.

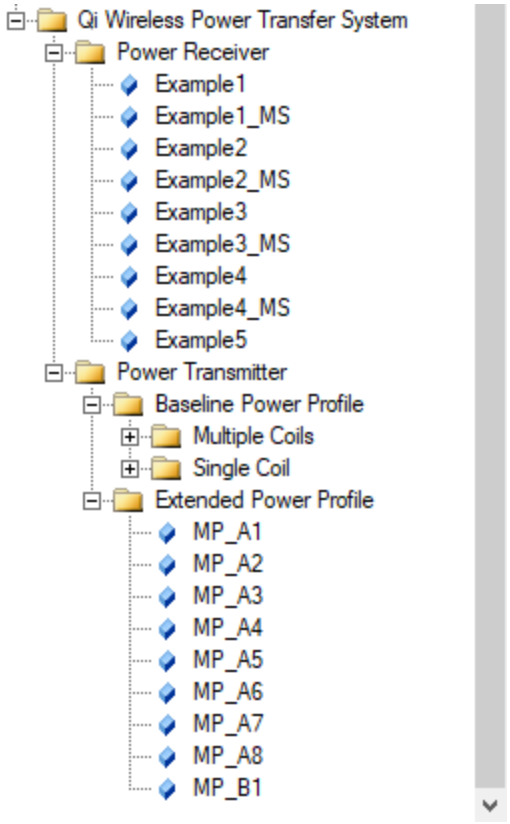
If you want to edit a 3D Component:

1. From Windows Explorer, copy and save the component design file to your PersonalLib or UserLib directory.
2. Do an update menu for 3D Components.
3. Open the component from the PersonalLib or UserLib directory and use edit mode for making modifications.

Qi Wireless Power Transfer System

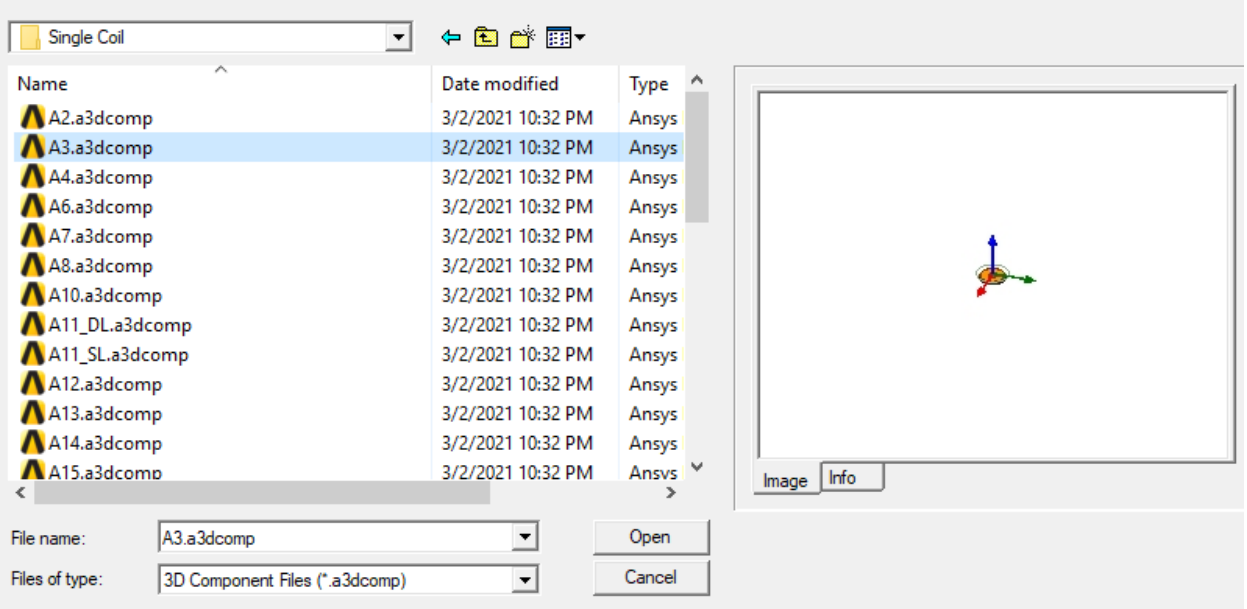
> **Note:** The Ansys models included in the Qi Wireless Power Transfer System 3D Component library are based on model specifications published by the *Wireless Power Consortium*.

There are two ways to view the Qi Wireless Power Transfer System components in the 3D Component Libraries. If you have enabled **View** > **Component Libraries**, you can navigate to the different models in the tree. From the **Component Libraries** tree, you can drag and drop models to the **Modeler** window.

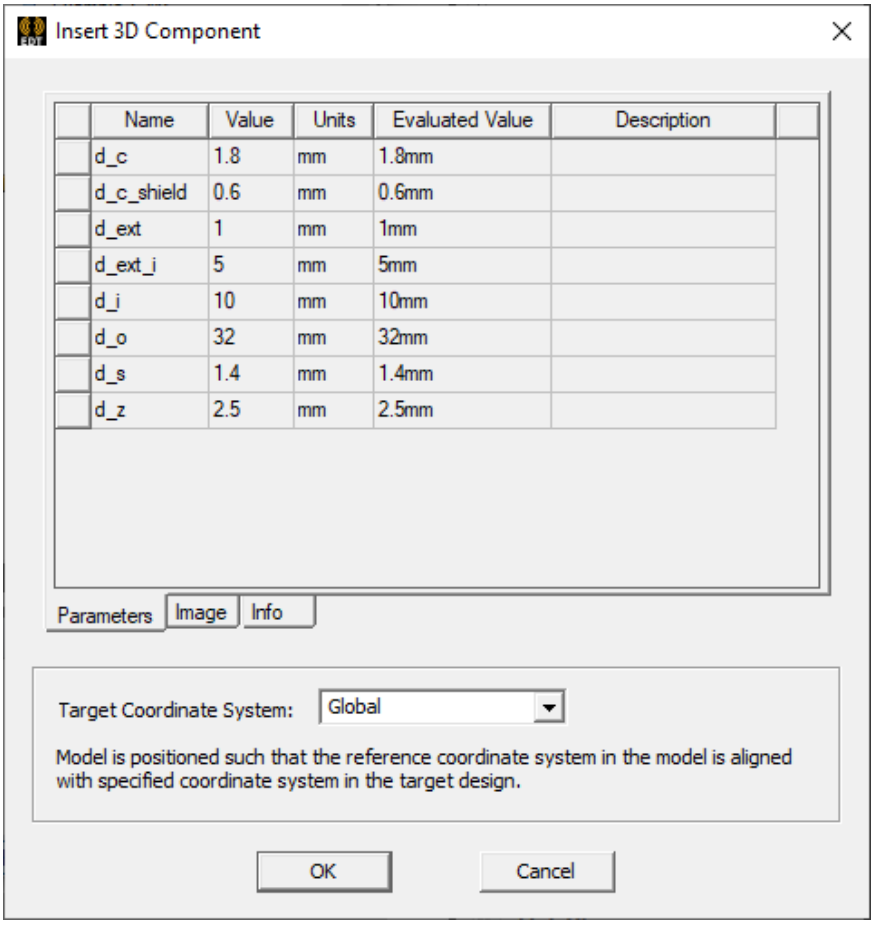


You can also access the **Qi Wireless Power Transfer System** library using **Draw** > **3D Component Library** > **Browse...** to open the **Browse 3D Component** dialog. You then click **SysLib** to display the libraries included in your installation.

You can select from the Qi Wireless Power Transfer System models, and navigate the folders to select from available components. Once you have selected a Component file, you can view the Image and Info for that component.



You can click **Open** to display the **Insert 3D Component** dialog box. You can view the Parameters, Image, and Info tabs for that component.



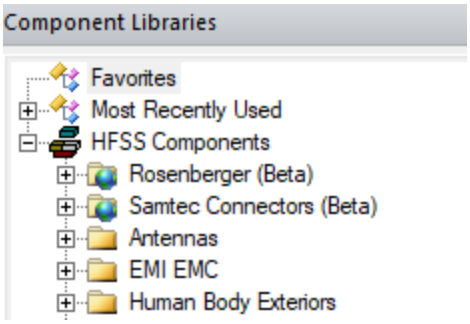
You can also open the **Insert 3D Component** dialog by double-clicking on a model in the **Component Libraries** window. (Refer to Inserting a Component into a Design.)

If you want to edit a 3D Component:

1. From Windows Explorer, copy and save the component design file to your **PersonalLib** or **UserLib** directory.
2. Open the component from the **PersonalLib** or **UserLib** directory and edit the component. (Refer to *Editing a 3D Component Definition*.)

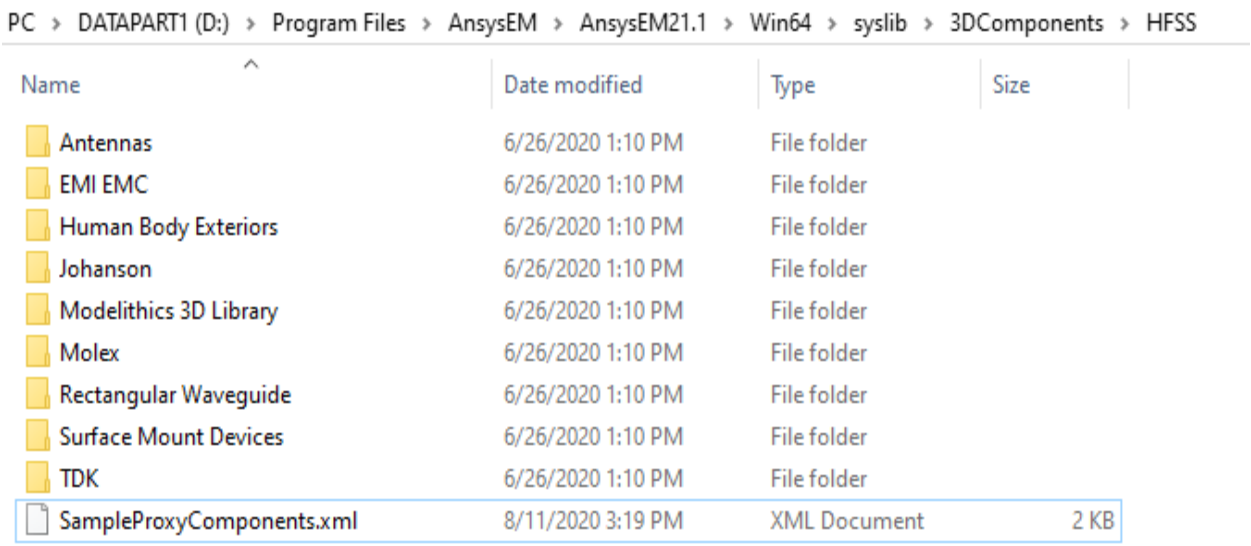
3D Component Agents: 3D Component Library

3D Component Agents is a feature that allows vendors to provide the information about 3D components that includes a link to their website for downloads. Such 3D Component Agents appear in the component browser along with other installed components, but shows them in a different color and with a globe icon or provide some other visual clue to indicate that these are different from components provided in the Ansys Electronics Desktop installation.



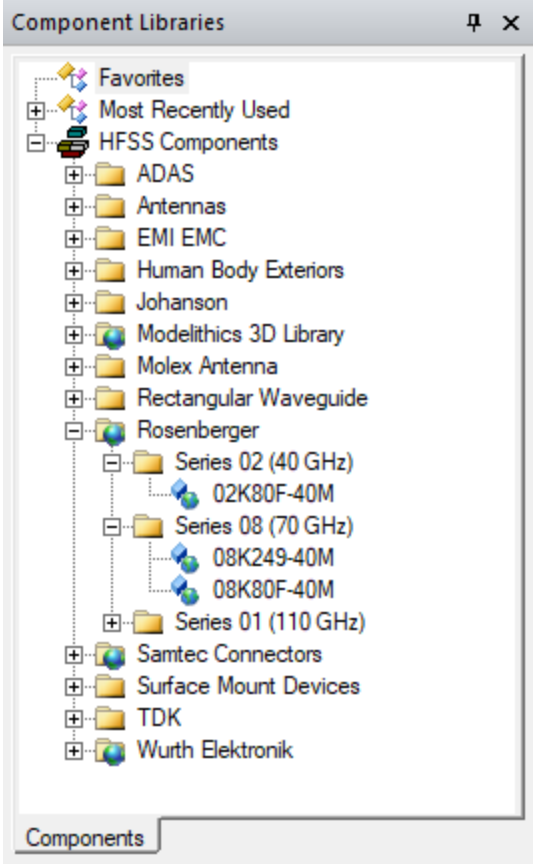
When you double click these 3D Component Agents, your default browser opens to a page with information about downloading the component. Dragging the proxy component into the editor window will do the same thing.

To view any 3D Component Agents, you must have a vendor-provided .xml file in the appropriate model directory:



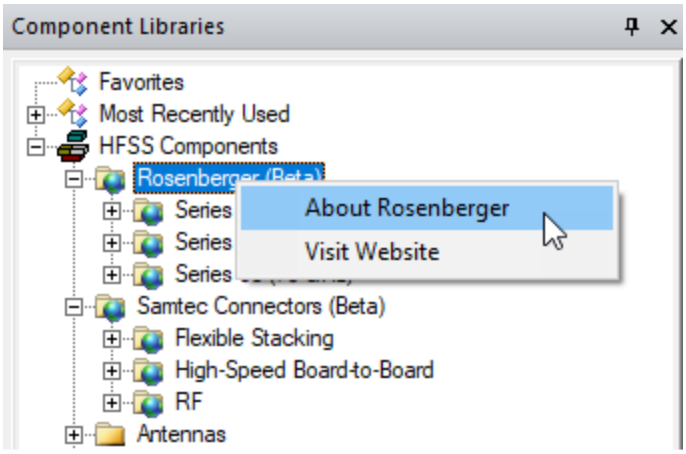
You can view 3D Component Agents in the 3D Component Library if you have enabled **View** > **Component Libraries**. You can see the **Component Libraries** window which will show available proxy libraries. Notice that folders for 3D Component Agents libraries have a small globe image overlaid on them, and the Agent components have a double icon (rectangle and circle) before the

name, rather than just a single blue rectangle.



Vendor Nodes of 3D Components Agent Libraries

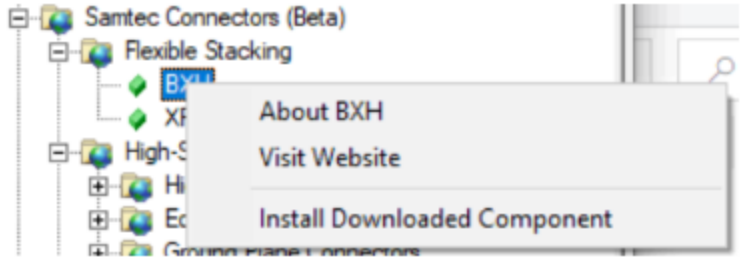
The vendors provide the name and other additional information such as part number, image, and so forth, for the 3D Components Agents, including the link to their website, and guidance for download, install, and organize them in a hierarchical folder structure just as the 3D components that are part of the installation libraries are organized. If you right click on the top node of a 3D Components Agent library, a menu lets you view information or visit their website.



- **About**: displays information about the whole group of components
- **Visit Website**: launch browser to website for whole group of components

3D Components Agent Node in Vendor Libraries

These nodes represent an individual proxy component.



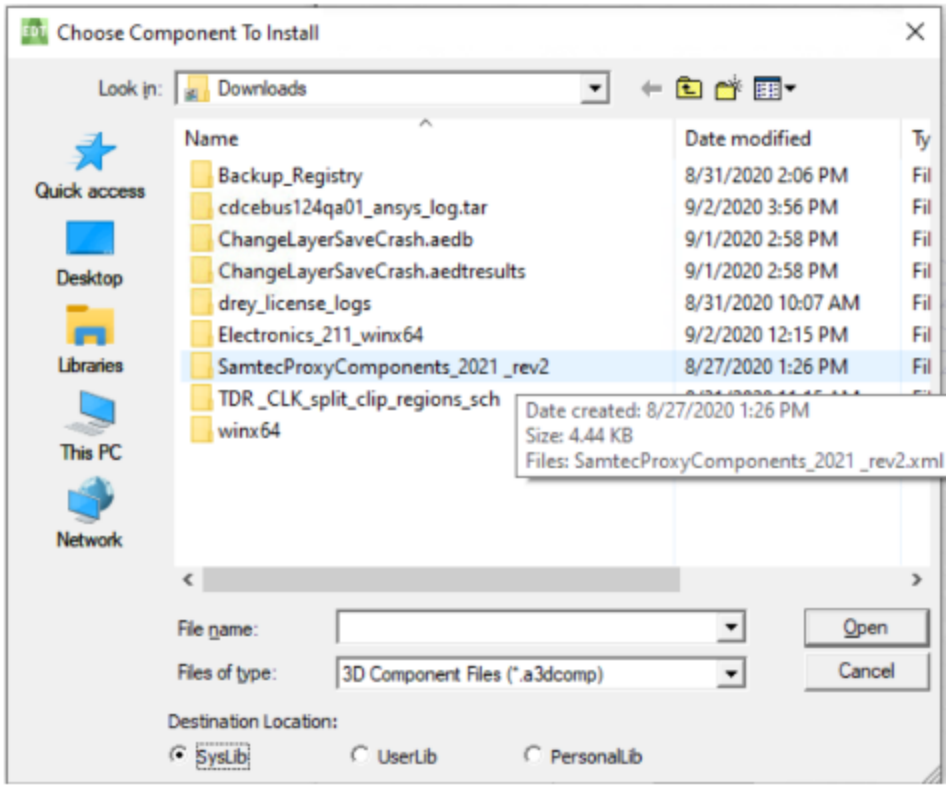
Right clicking on a component displays a menu with three items:

- **About**: displays information about the selected component
- **Visit Website**: launches browser to view information about the selected component, and possibly purchase/download the selected component.
- **Install Downloaded Component**: allows user to import the downloaded component into the component browser.

Installing Downloaded Components

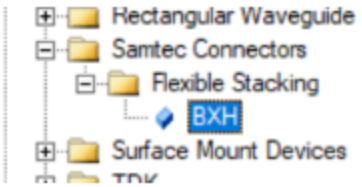
Once an agent has been purchased/downloaded, it needs to be installed into your component library. Right click on the Agent component node corresponding to the component you downloaded and choose **Install Downloaded Component**. Note: it is VERY important that you select the correct component node that corresponds to the component file that you downloaded or the component file may be installed as the wrong component.

When you click **Install Downloaded Component**, you will be shown a file browse dialog where you can choose the purchased/downloaded component file. See below.



You can also choose to import the component into your syslib, userlib, or personal directory.

Once the component file has been installed, it will now show up as a normal component that can be inserted into the design. See below.



You can drag and drop installed models from the **Component Libraries** window to the **Modeler** window.

Legacy Projects with 3D Components

Components created in earlier projects do not have the same degree of encapsulization and are not compatible with the current version of Maxwell.

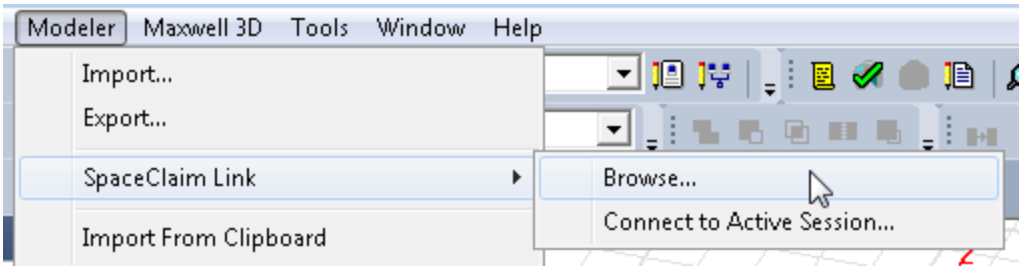
SpaceClaim Integration

Ansys Electronics Desktop can import geometry from a SpaceClaim document using the **SpaceClaim Link** feature. SpaceClaim Link can be used: if the SpaceClaim Direct Modeler (SCDM) is part of an Ansys installation that is of the same version as Ansys Electronics Desktop; or if a standalone installation of the SpaceClaim modeler exists on the same machine.

SpaceClaim Link is only available on Windows, and provides the following functionality:

Ansys Electronics Desktop can import geometry from the SpaceClaim modeler. To use **SpaceClaim Link** with Ansys Electronics Desktop, you must install the SpaceClaim modeler on the same Windows machine. SpaceClaim Link provides following functionality:

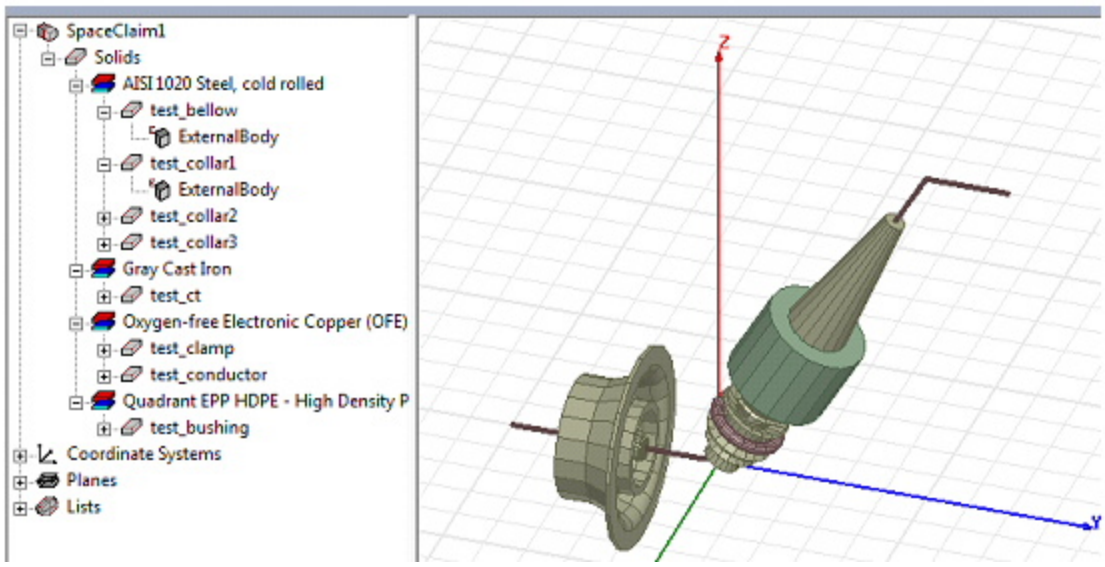
- Browse and select SpaceClaim documents to import geometry directly.
- Import geometry by connecting to a active SpaceClaim session.
- The SpaceClaim geometry is imported as a User Defined Model (UDM) along with geometry parameters with units, rendering attributes, and material assignments.
- For a SpaceClaim link for a UDM already in the design, you can launch a SpaceClaim session from AEDT and open the corresponding SpaceClaim document.
- A SpaceClaim document referenced by a SpaceClaim link UDM is included in project archive automatically.
- By option you can break a connection to SpaceClaim. In this case, all UDM parts are then converted to imported objects. This feature lets you use SpaceClaim just for its healing capacity.



Importing SpaceClaim geometry into Ansys Electronics Desktop

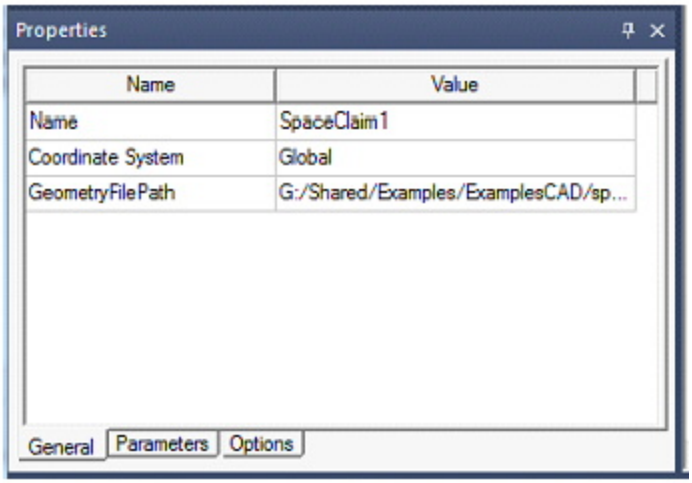
1. Click Modeler>SpaceClaim Link>Browse to open a file browse dialog.
2. Select a SpaceClaim .scdoc document file and import the geometry.

The imported geometry is shown in the History tree as a User Defined Model (UDM).

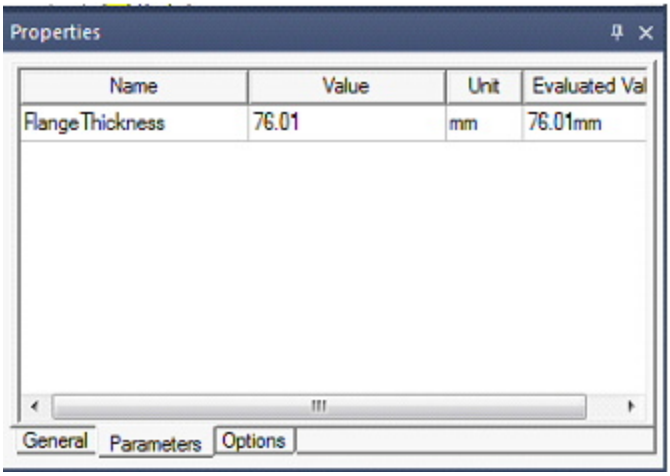


The Properties window of a SpaceClaim link UDM has 3 tabs.

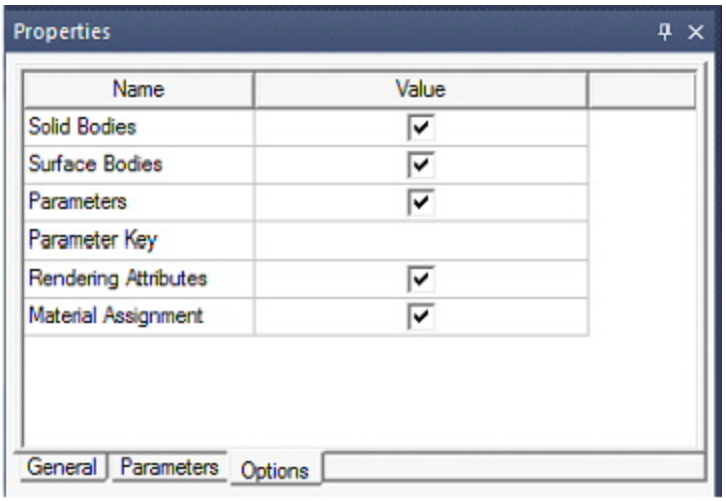
- The **General** tab shows general information such as SpaceClaim document path.



- The **Parameters** tab shows all parameters.



- The **Options** tab shows supported options.



Options Include:

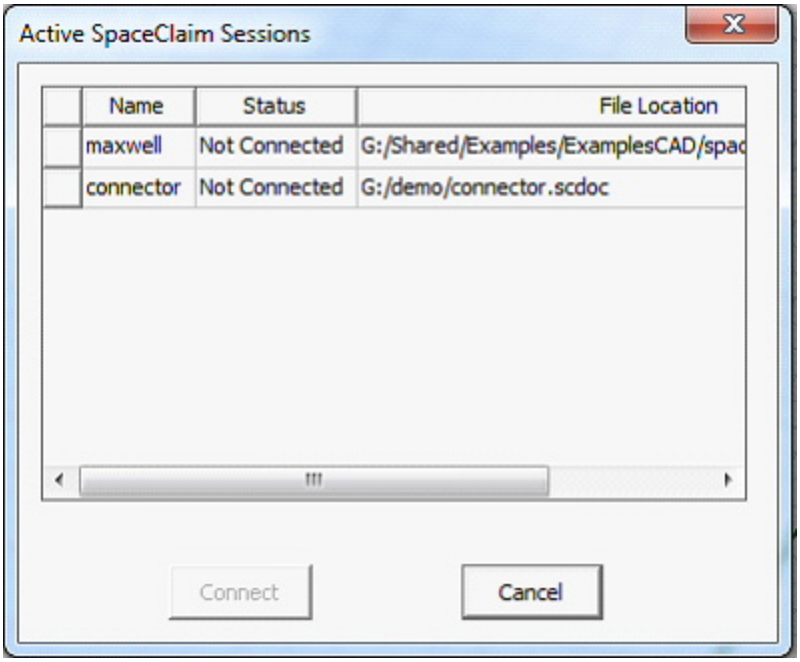
- **Import Solid Bodies**: Default is ON
- **Import Surface Bodies**: Default is ON
- **Impost Parameters**: Default is ON (It imports parameters).
- **Parameter Key**: A string that is used to filter out parameters. Default is no string - i.e. all SpaceClaim parameters will be imported.
- **Impost Rendering Attributes**: We import color of objects. By default the option is ON.
- **Import Material Assignments**: By default we import material assignment. Note that material properties are not imported. A material with exact name must already exist. Otherwise, validation issues a violation.

- **Import suppressed for physics objects**: By default, this is unchecked and objects suppressed for physics are not imported. If you check Import suppressed for physics objects option in the Property window for a SpaceClaim object, and you refresh the SpaceClaim link by "Send Params and Generate", suppressed objects are imported as non-model objects. Once imported, you can change them to model objects if desired.

Connecting to an Active SpaceClaim session:

If one or more SpaceClaim sessions are running on the machine, **SpaceClaim Link** can connect to them to import geometry directly.

When you click the **Connect to Active Session** menu item, Ansys Electronics Desktop opens a dialog showing all active SpaceClaim sessions, their document name and status.

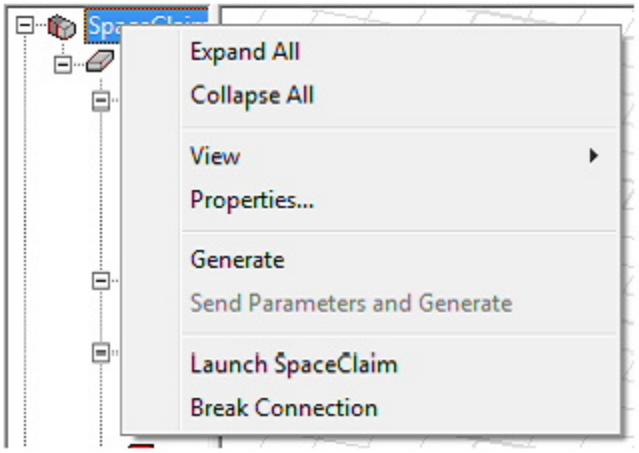


You can select a session with “Not Connected” status and press the **Connect** button to import geometry from that session. Once connected, that session is used to regenerate the corresponding SpaceClaim link UDM.

The dialog is empty if no sessions are running.

Launching a SpacClaim session from Ansys Electronics Desktop

The context menu of a SpaceClaim linked UDM shown in the History tree allows you to **Launch SpaceClaim** and to **Break Connection**.



When you select **Launch SpaceClaim**:

- If no SpaceClaim session is open, a new SpaceClaim session is opened, and the document corresponding to SpaceClaim link will be opened in it.
- If a SpaceClaim session is already open, the document will be opened in that session.

All refresh/regenerates of the linked SpaceClaim document use the session that is opened.

The session closes when Ansys Electronics Desktop closes.

Breaking Connection with SpaceClaim Link:

When you select the **Break Connection** command in the context menu of the SpaceClaim link UDM:

- Break Connection converts the UDM bodies to imported objects. The UDM will be removed.
- Any operation that you had performed on UDM bodies is preserved.
- Parameters of the UDM are lost. You lose the ability to create parametric variations with SpaceClaim parameters. Note that parameters for subsequent history operations on SpaceClaim parts will be retained.
- Attributes such as rendering attributes, materials, etc., are preserved.

Archive a Project containing SpaceClaim link

If you want to share a project with a SpaceClaim linked document with other users, you need to archive the project. Sharing the project file only is not sufficient because the SpaceClaim document is not included in the project.

- SpaceClaim documents corresponding to the link will be included in the project archive automatically.

- If any SpaceClaim document refers to other documents (e.g., an assembly document might refer to individual part documents), these indirectly referenced documents will also be included in the Ansys Electronics Desktop project archive.
- When an archive is unarchived, referenced SpaceClaim documents will be put in a “restored_files” subfolder in the project directory. All definition paths are updated to refer to the reference files in the “restored_files” folder.

Handling SpaceClaim document Save

When you launch a SpaceClaim session from Ansys Electronics Desktop or connect to a live session, whenever the project is saved, the SpaceClaim document corresponding to the SpaceClaim link UDM is automatically saved if any unsaved changes exist in that document.

If unsaved changes in SpaceClaim were not imported into Ansys Electronics Desktop before saving, they will be imported the next time that the SpaceClaim link UDM is regenerated.

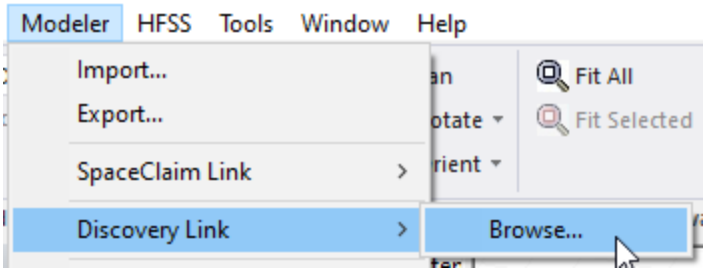
You must ensure that the geometry in Ansys Electronics Desktop and that in SpaceClaim are in sync.

Related Topics

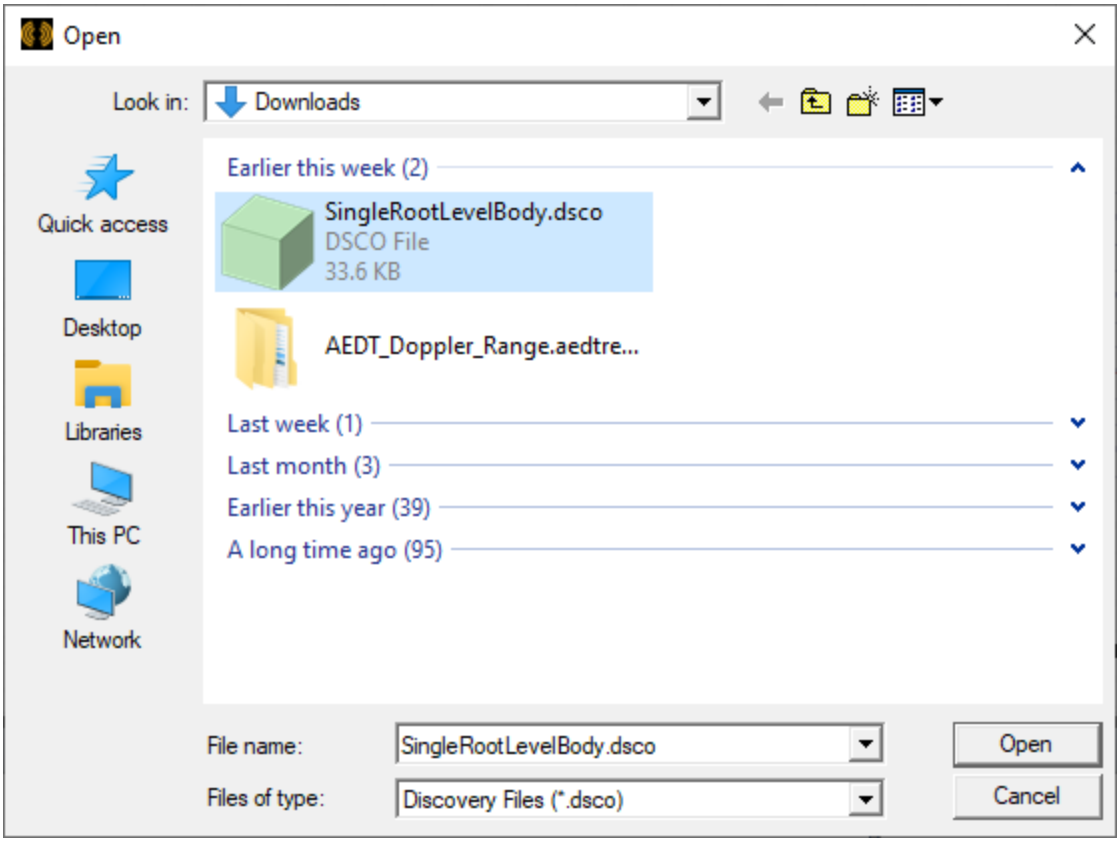
"User Defined Model (UDM)" on page 18-36

3D Modeler Discovery Link

The Beta feature 3D Modeler Discovery Link lets select **Modeler>Discovery Link>Browse...** to open a browser window to select a dsco project and create a user defined model from the geometry model used in the dsco project. After that, you can select the user defined model and "Break connection" to make the UDM objects become import objects. IDs are not preserved in Beta release.



You can select a *.dsco project to load into the Modeler window.



Click **Open** to create a user defined model from the geometry model used in the dsco project. After that, you can select the user defined model and "Break connection" to make the UDM objects become import objects. IDs are not preserved in Beta release.

Related Topics:

User Defined Model (UDM)

Model Analysis

For some models it may be beneficial to remove unnecessary small entities and to fix object misalignments to avoid potential mesh issues. Maxwell includes Model Analysis functions to help you evaluate models you have imported or created. Select **Modeler> Model Analysis** to see the menu options. Depending on the design and the current selection, some features may not be enabled. The menu includes the following commands.

- Analyze Objects
- Analyze Interobject Misalignment
- Analyze Surface Mesh
- Heal
- Show Analysis Dialog
- Align Faces

- Remove Faces
- Remove Edges

Note	Before running model analysis, you must remove all command history for the selected object by using the Purge History command. If you need to save the object history, save a separate copy.

1. After import, you typically perform validation check. This lets you focus on objects and object pairs that have errors and or warnings. The objects that fail should be analyzed by using the **Modeler>Model Analysis>Analyze Objects** menu item.
2. Select the objects and invoke **Modeler>Model Analysis>Analyze Objects**.

   This displays the **Analysis Options** dialog to allow you to specify settings for entity check level, and small feature detection.

   When you **OK** this dialog box, the initial analysis executes and the **Model Analysis** dialog is displayed.
3. Choose the objects that have "Invalid Entities Found" and **Perform>Heal Objects**.

   In most cases, the objects will be healed and the errors fixed.
4. If errors still persist, choose the edges and faces and click on **Delete**.

   This will replace the selected face/edge object by a tolerant edge/vertex respectively. In some cases the replacement of the face/edge by tolerant edge/vertex will fail.

When models pass the initial validity checks, mesh generation could still fail. The following errors can be present in models: (Refer to Detecting Errors.)

1. Non-manifold topology. These are non-manifold edges and vertices that are present in the model.

Note	One of the requirements for valid mesh generation is that the surface mesh must be ***manifold***, that is, a plot of all the surface triangles of a 3D solid object should not have any “holes.” Such a mesh is often referred to as a ***watertight*** mesh. If there is a hole and you were to fill the object with water, the water would leak out of that hole – the mesh would not be ***watertight***. Zero-area holes (often called cracks) will also “leak.” Having multiple triangles cover the same patch also results in a mesh that is not manifold. Refer to *Healing Models*, and the list of **Related Topics** below for information on detecting and correcting such errors.
2. Object pair intersection. This detects whether pairs of objects intersect.
3. Small feature detection – small edge length, small face area and sliver face detection.
4. Mis-aligned entities detection – detects pairs of faces from objects that can be aligned to remove object intersections. This improves the probability of mesh success.
5. Mesh failure error display. This is available for single object, object pairs and last simulation run (all objects in a model). Errors reported by the meshing module are reported to the user.

Errors of type 3 and 4 should be resolved before you invoke the meshing for the model.

By default, the **Heal** command is automatically applied to imported objects.

Related Topics

Set Material Override
Analysis Options Dialog
Healing
Healing Non-manifold Objects
Healing Options
Technical Notes: Healing Models
Technical Notes: Healing and Meshing
Technical Notes:*Detecting and Addressing Model Problems to Improve Meshing*

Analysis Options Dialog

To perform analysis on an object according to specified features and tolerance values:

1. Select the object you want to analyze and click **Modeler> Model Analysis>Analyze Objects.**

   This displays the **Analysis Options** dialog, with the **Analysis Options** tab selected. Selecting **Modeler>Model Analysis>Heal** also displays this dialog box. If, during **Modeler>Import...** you select **Heal Imported Objects** and **Manual** on the file browser dialog box, you also see this dialog.
2. If desired, check the **Perform Entity Check Errors** check box.

   This enables the **Check Level** menu. The setting can be **Basic**, **Strict**, or **Comprehensive**.
3. If desired, click the check boxes to enable and set the **Detect Feature** settings:
   - **Detect Holes**, and specify the Maximum Radius.
   - **Detect Chamfers**, and specify the Maximum Width.
   - **Detect Blends**, and specify the Maximum Radius.
4. If desired set the **Detect Small Entities** features and tolerance values.
   - **Small Edges**, length less than
   - **Small Faces**, area less than
   - **Sliver Faces**, which enables:
     - Object Bounding Box Scale Factor
     - Sliver Edge Width
5. Click the **Properties** tab to see a listing of the geometric properties of the selected object.
6. Clicking **OK** on this dialog displays the **Model Analysis** dialog which contains the results of the analysis.

Related Topics

Healing an Imported Object

Model Analysis Dialog
Set Material Override

Analyzing the Surface Mesh

To set the options to analyze the surface mesh:

1. Select an object of interest.

   This enables the Analyze Surface Mesh command in the menu.
2. Click **Modeler>Model Analysis>Analyze Surface Mesh**.

   The **Surface Mesh Analysis Options** dialog box appears. This dialog box allows you to set parameters to remove.
   - You can also open the **Surface Mesh Analysis Options** dialog box from the **Model Analysis** dialog box via the **Perform** pull-down menu on the **Objects** tab.
3. Select or clear the **Perform Object Pairs Analysis** check box. Selecting this option evaluates the mesh for all combinations of the selected objects.
4. Select or clear the **Ignore Objects Separated by greater than** check box, and enter a value in the text box. Selecting this option means that object pairs are disregarded from analysis if their separation is greater than the specified value.
5. Click **OK** to perform the analysis with the selected options.

   The **Model Analysis** dialog box appears, displaying the results of the analysis.

Related Topics

Healing an Imported Object
Viewing Model Analysis Results (Model Analysis Dialog Box)

Model Analysis Dialog

This dialog contains results for all model analysis, including diagnostic information relating to mesh issues. To view the analysis options:

1. Click **Modeler> Model Analysis>Show Analysis Dialog**.
   A submenu appears.
2. Select one of the following from the submenu:
   - **Objects**
   - **Objects Misalignment**
   - **Surface Mesh**
   - **View Mesh Feedback**

   The **Model Analysis** dialog box appears. (This dialog box also appears automatically after clicking **OK** in the **Analysis Options** dialog box.)

3. You can select **Hide All Other Objects** to suppress display of objects, faces, or surfaces in the Modeler window not selected in the **Mesh Feedback** panel
4. Select the **Auto zoom to selection** check box to automatically zoom to the item selected on the **Objects** tab.
5. Make the desired changes on each tab in the **Model Analysis** dialog box :
   - **Objects** tab
   - **Objects Misalignment** tab
   - **Surface Mesh (Single/Pairs)** tab
   - **Mesh Feedback** tab
5. Click **Close** to close the **Model Analysis** dialog box.

Related Topics

Healing an Imported Object

Analyzing Objects

Analyzing the Surface Mesh

Set Material Override

Technical Notes*: Healing and Meshing*

Technical Notes: *Detecting and Addressing Model Problems to Improve Meshing*

Objects Tab

All results relating to model analysis of specific objects are presented under the **Objects** tab. The **Objects** tab for the **Model Analysis** dialog lets you view the following information:

1. The results table contains the following information.
   - **Name** – column listing the objects in the current design.
   - **Last Analysis status** – column giving the analysis status of the listed objects. Objects can have the following status:
     - **Good** – the object contains no invalid geometry entities given the tolerance values specified in the **Analysis Options** dialog.
     - **Null Body** – the object is non-existent.
     - **Analysis not performed** – the object was not selected for analysis.
     - **Invalid entity errors** – these are api_check_entity() errors and non-manifold errors which must be fixed prior to meshing.
     - **Small entity errors** – small faces, sliver faces and small edges that are optionally detected based on the tolerance limits specified in the **Analysis Options** dialog.
2. Select any object name in the table which contains errors to display a set of radio buttons in the panel and a list of corresponding faces, edges and vertices.

Note	**Auto Zoom to Selection** – if this option is checked, the modeler automatically zooms to the item selected in the **Model Analysis** dialog box.

3. Select the face, edge or vertex entity from the list to view the error description in the **Description** field.
4. Select the **Delete** button if you want to remove a selected face or edge entity.
5. Select the **Perform** button to list the commands that you can execute on the selected objects in the Results table.
   - **Heal Objects -** repairs invalid geometry entities for the selected objects within the specified tolerance settings. The **Healing Analysis** dialog will appear.
   - **Analyze Objects** – evaluates the object status. Selecting this displays the Analysis Options dialog.
   - **Analyze Surface Mesh** – invokes a mesh for each selected object and reports analysis results under the **Surface Mesh (Single/Pairs)** tab. Selecting this option displays a dialog with radio buttons to select.
     - Perform Object Pairs Analysis - evaluates mesh for all combinations of the selected objects.
     - Ignore objects separated by greater than a specified value - object pairs are disregarded from analysis if their separation is greater than the specified value.
     - Click **OK** to perform the analysis with the selected options.
   - **Analyze Interobject Misalignment** – determines any misalignments between two selected objects in the results table. The results are reported under the **Objects Misalignment** tab.
   - **Display Healing Log** – checking this causes the **Model Analysis** dialog to display a healing log which includes information about operations performed on an object during the healing process.

Related Topics

Analyze Objects

Analyze Interobject Misalignment

Analyze Surface Mesh

Healing

Set Material Override

Objects Misalignment Tab

The table in this panel displays results of an **Interobject Misalignment** analysis. All misaligned face pairs corresponding to the analyzed objects are listed in the table.

- **Align Faces** – select a face pair in the table and click the **Align Faces** button to align selected faces.
- **Clear All Analysis Data** - this button removes all information from the tables.
- **Auto Zoom to Selection** – if this option is checked, the modeler automatically zooms to the item selected in the table.

After validation check is performed, the pairs of objects that intersect are chosen for analysis. Use the analysis results to find whether objects have faces that can be aligned.

Choose all the bodies that intersect with another body.

1. From the **Model Analysis** dialog choose perform/Analyze Interobject misalignment. Or you can run **Modeler>Model Analysis>Analyze Interobject Misalignment**.

   If the analysis finds object pairs that can be aligned, they will be displayed in the **Objects Misalignment** tab.
2. You can select individual or multiple rows and perform Align Faces. In some cases, face alignment will fail if the topology of the body changes by a large factor after alignment.
3. Identify individual bodies and body pairs that fail to mesh.
4. Perform Mesh analysis on individual objects and object pairs.
5. Review the reports and fix the errors.

Related Topics

Analyze Objects

Analyze Interobject Misalignment

Analyze Surface Mesh

Healing

Technical Notes: *Healing and Meshing*

Set Material Override

Surface Mesh (Single/Pairs) Tab

The panel displays the results of a surface mesh analysis.

1. You can display results for:
   - Individual Objects
   - Object Pairs

Note	**Auto Zoom to Selection** – if this option is checked, the modeler automatically zooms to the object or object pair selected.
2. The results table contains the following information:
   - Object - column listing object name or a pair of object names.
   - Last Analysis Status - column stating the meshing status of the object or object pair.
     - Mesh Success
     - Mesh Failure
   - Error Type - this column gives the category of error that caused the mesh failure.
   - Error Detail - provide specific geometry information regarding mesh error location.

   Display options include:

- **Display Mesh Analysis** log check box -checking this displays further details concerning each error to be listed.
- **Auto Zoom to Selection** – checking this causes the modeler to automatically zoom to objects or faces corresponding to the error.

Related Topics

Analyze Objects

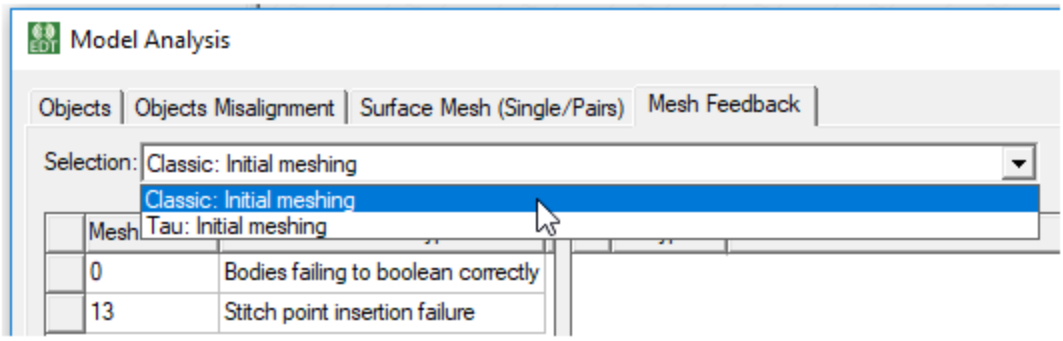
Analyze Interobject Misalignment

Analyze Surface Mesh

Healing

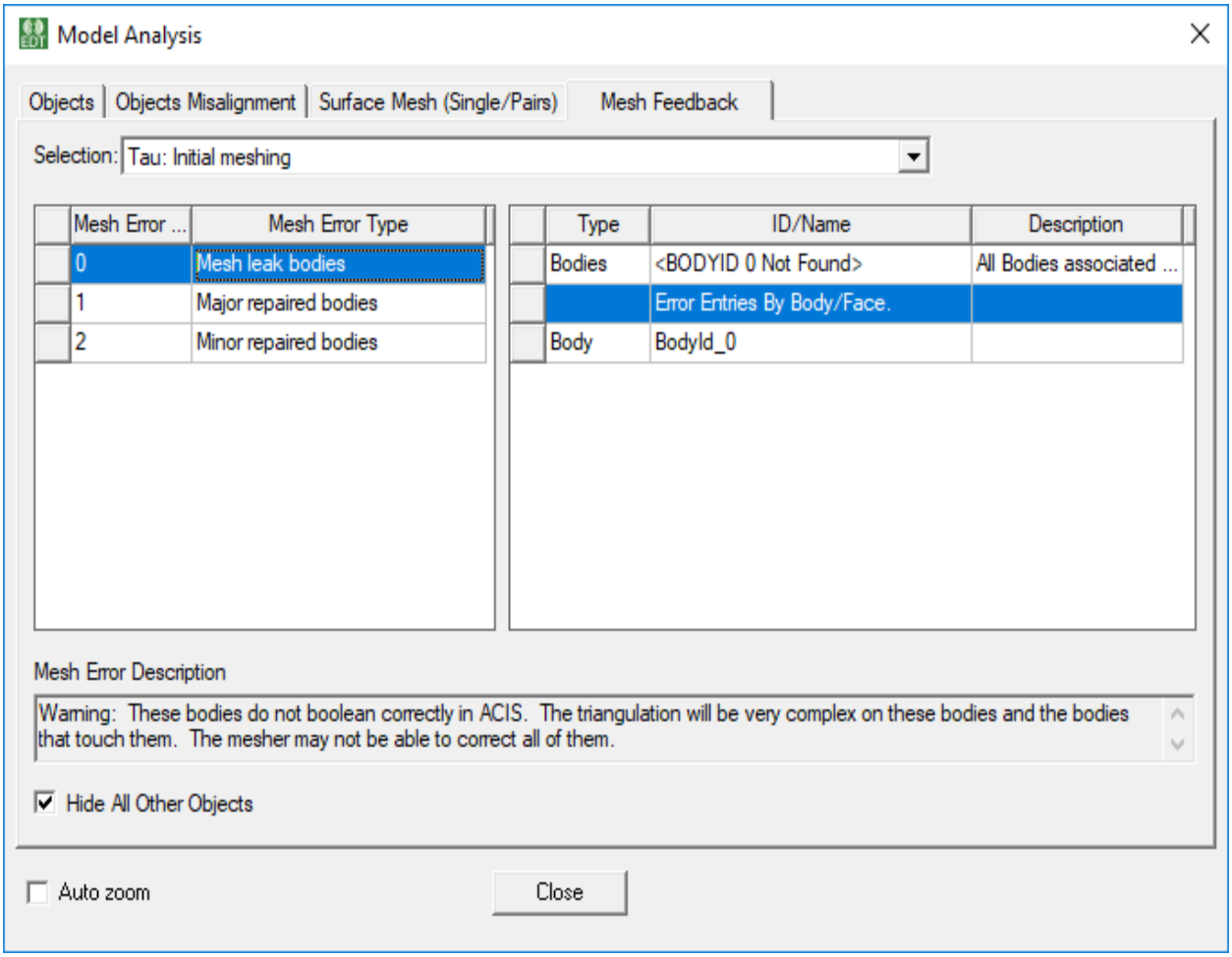
Mesh Feedback Tab

The tables in this panel lists all model errors as viewed by the mesher. If the project has solutions by different solvers, the **Selection** drop down lets yous select which to view.



The left side table has columns for Mesh Error Serial ID and Error Type.

- Mesh Error Serial ID - the 0 item is the body with which the subsequently listed entries are associated.
- Error Type - this column gives the category of error that caused the mesh failure, for example, Non-Manifold Edge, or Point Insertion Failure.



The display in the table on the right side of the **Model Analysis** dialog box depends on your selections from the left hand table. When you select a row in errors panel (the left panel as shown in above figure), all the faces are selected (second row in right panel) and if you enable **Auto Zoom**, the view fits to the errors:

- Type - whether the error applies to Bodies, Faces, or Triangles. If you select a particular error type, the columns for Type, ID/Name and Description are filled in.
- ID/Name -- the object name or object ID for the error type.
- Description -- the first three rows are for Bodies, Faces, and mesh Triangles. Subsequent rows, are of Error Entities By Body.Face.

The **Mesh Error Description** field describes error message describing the nature of the selected Mesh Error Serial ID, the implications, and provides a recommended response.

Display options include:

- **Display Mesh Analysis** log check box -checking this displays further details concerning each error to be listed.
- **Auto Zoom to Selection** – checking this causes the modeler to automatically zoom to objects or faces corresponding to the error.

Use of these selections let you more easily view and respond to the errors. If you enable **Auto Zoom**, any selection whether it is body(ies), face(s) or error triangle(s) or error segment(s) will fit to view. If all the faces are also selected (second row in right panel), then errors and selected entities will fit to view.

Focusing only on the errors when an error is selected in right panel, improves visualization of these meshing errors and provides you with precise information in order to take corrective action quickly.

Mesh Error Markers

As shown above the mesh errors are marked by red circles. The following table describes all such markers:

	Mesh errors
	Mesh warnings
	Mesher information
	Selected Faces

These markers are located at the center of errors or center of faces and their size correspond to the maximum side size of the entity, such as error triangles, error segments or faces. The size of these markers is clamped to a maximum and a minimum size. Additionally they have some transparency. The size and transparency of these markers cannot be changed by the user.

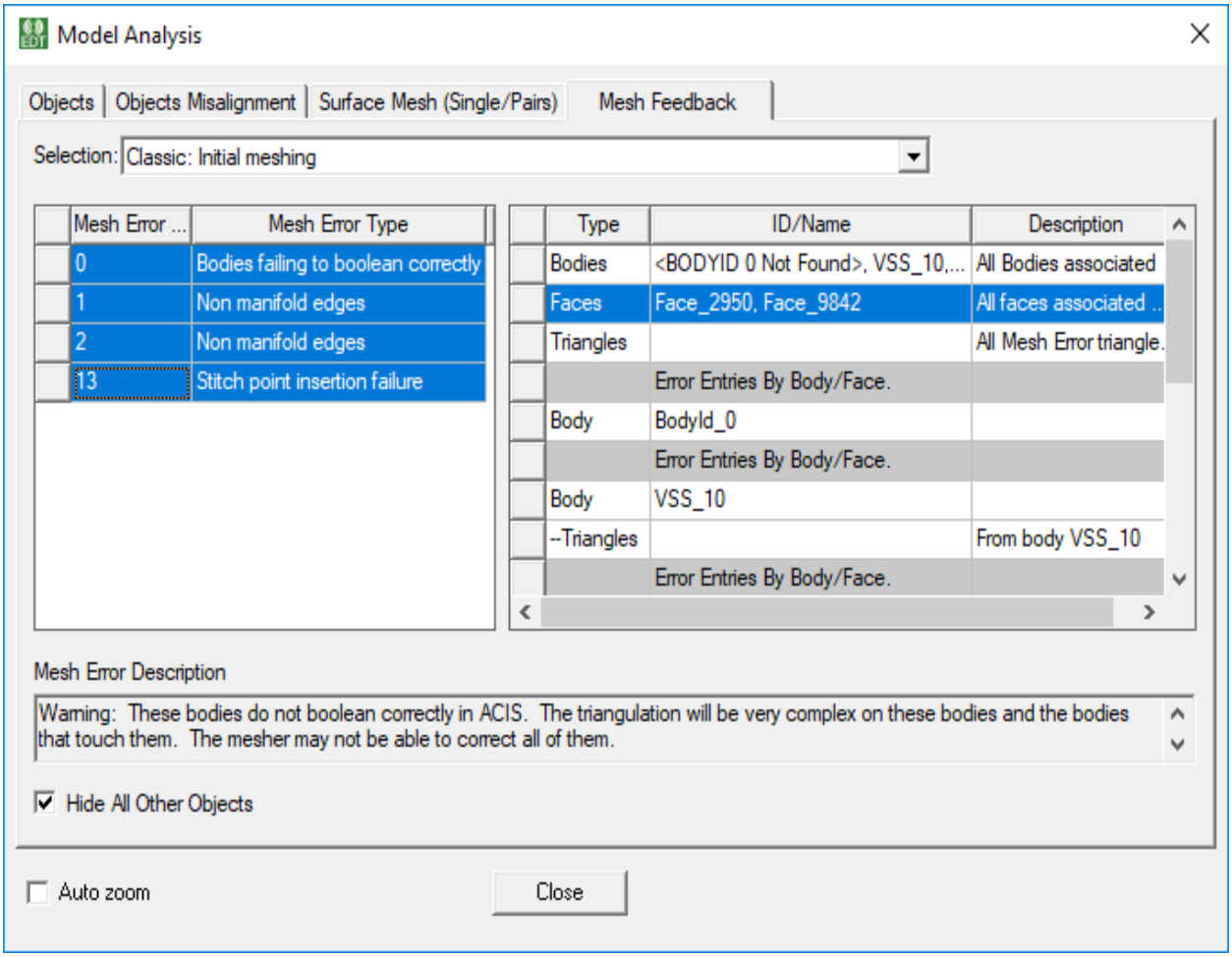
If many individual faces are selected in the right panel, each face is marked by a blue square. To find which blue square corresponds to which face in the right panel in the dialog box, click on the blue square. This action selects the corresponding face row in the right panel of the dialog box.

Face Selection

When the all faces row or individual face rows are selected, the Modeler selects the face(s). This enables the face related commands in the Modeler window. Also when you select faces this way, it makes the corresponding bodies of the selected faces visible, even when **Hide all other objects** is enabled. This provides some contextual information about the face location with respect to its body.

Multiple Selection of Mesh Errors

You can select multiple rows in the right pane of the **Mesh Feedback** tab. To view all the meshing errors, select all the rows and enable **Auto Zoom**. With **Auto Zoom** enabled, the view fits to all the errors, providing an overview of all the mesh errors. Also, when multiple errors are selected, right panel shows the merged information about all bodies, faces, error triangles etc. The **Mesh Error Description** text box shows description of all the errors.



Related Topics

Analyze Objects
Assigning Skin Depth Based Mesh Refinement
Analyze Interobject Misalignment
Analyze Surface Mesh
Healing

Model Preparation Commands

Use the **Modeler>Model Preparation>** commands to prepare a model.
These commands include:

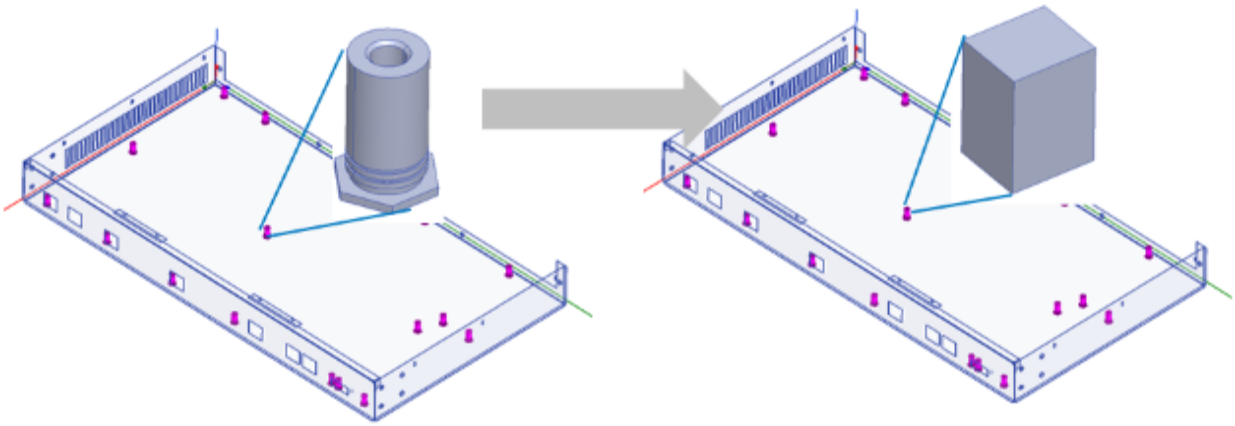
- Simplify
- Heal
- Stitch Sheets
- Align Faces
- Remove Faces
- Remove Edges
- Perform Explicit Subtractions

Related Topics

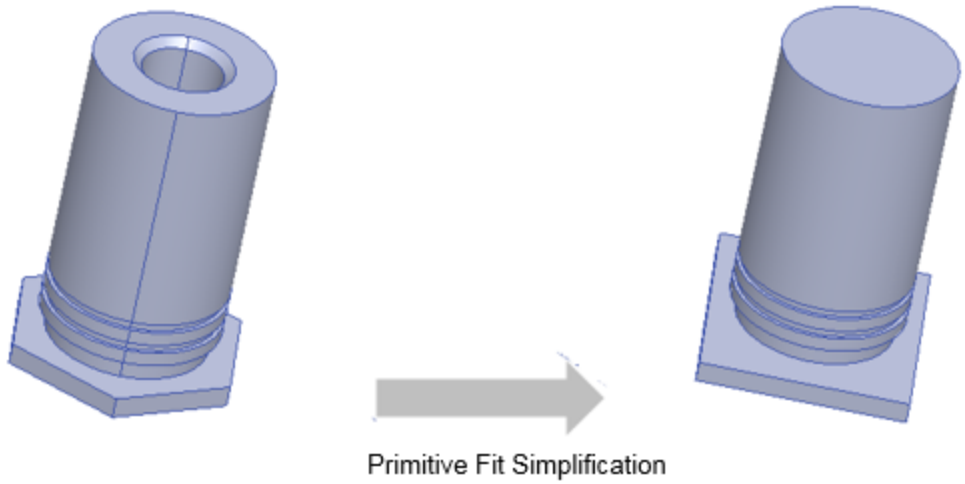
Model Analysis
Modifying Objects
Drawing Objects

Simplify Command

Use the **Modeler>Model Preparation>Simplify** command to converts a complex MCAD object into simpler primitives which are easy to mesh and solve. The operation can be applied on any selected object and not just imported objects. You can specify the type of simplification as Bounding Box, Primitive Fit, or Polygon Fit.



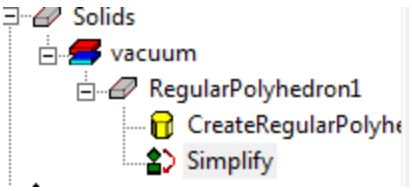
Bounding Box Simplification



Primitive Fit Simplification

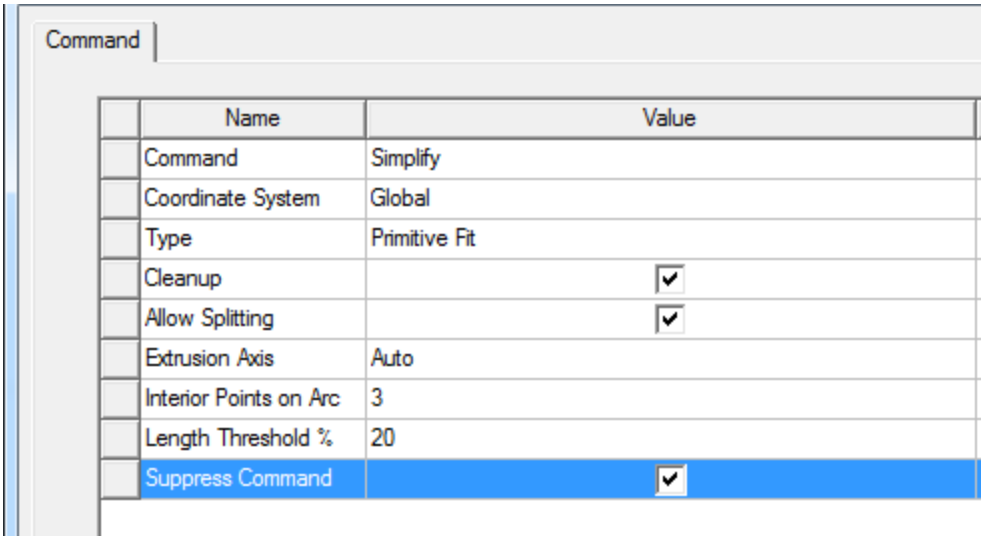
Polygon Fit Simplification

The History tree shows the **Simplify** command as being applied to the object.



Any parameters you specify in the dialog you can also edit in the Properties dialog for the History tree **Simplify** command to get a different simplification. The properties display in the docked properties includes only the parameters that are used by your currently selected **Simplify** command type. None of the parameters of simplify operation accept variables. If you right-click on Simplify for a project that was created for an earlier version, the menu displays and option to

Upgrade version. After an upgrade, the option not longer appears. The Properties dialog for the Simplify command allows you to suppress the command.



1. Select an object.

   This enables the Simplify command on the **Modeler>Model Preparation** submenu.

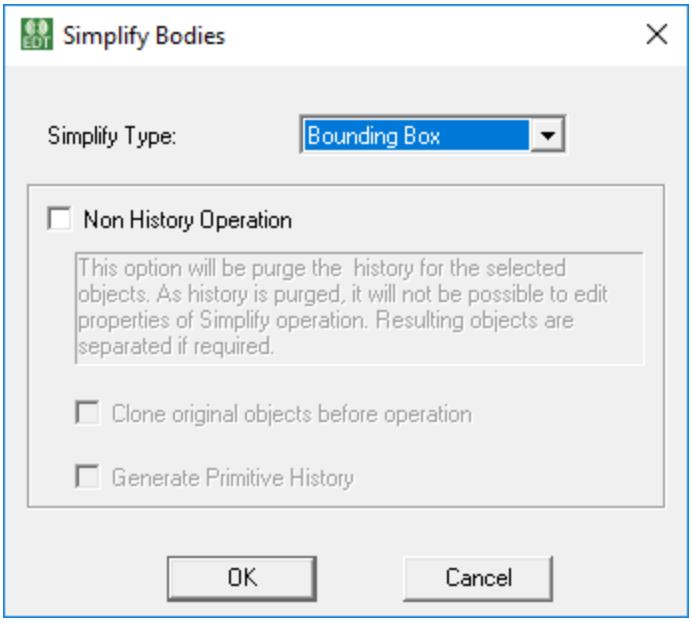
2. Click **Modeler>Model Preparation>Simplify**

   This opens the **Simplify Bodies** dialog box. A drop down menu lets you select the degree of simplification, from Bounding Box, Primitive Fit, or Polygon Fit.

   Bounding Box Simplify Type

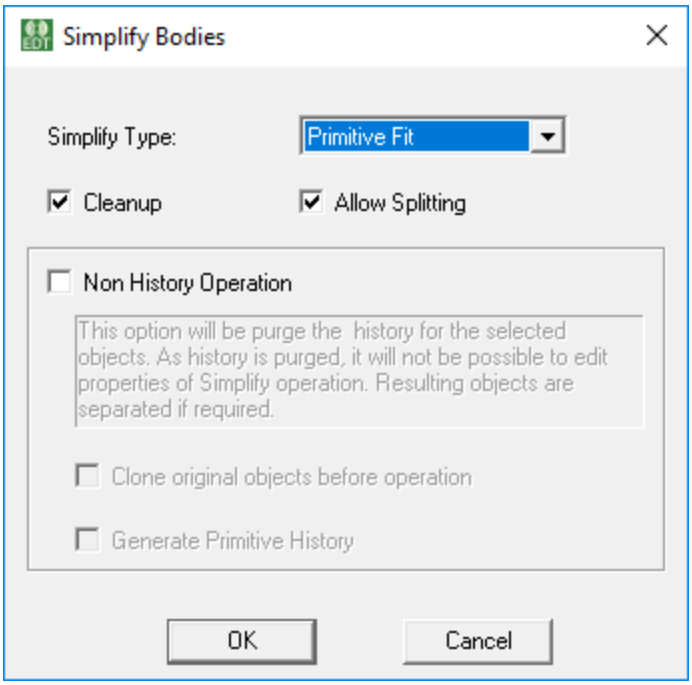
   If you select Bounding Box as the Simplify Type, there are no additional parameters. The object is replaced by its exact bounding box as computed in operation's coordinate system. You can change the operation CS to get a bounding box in an appropriate orientation.For

the **Non History Operation** options, see below.



Primitive Fit Simplification

For Primitive Fit simplification the object is replaced by a set of primitives shapes like Prism, Cylinder, Cone and so forth. This simplification type typically produces the simplest geometry with highest number of primitives. Primitive fit has the following options: Cleanup and Allow Splitting. For the **Non History Operation** options, see below.

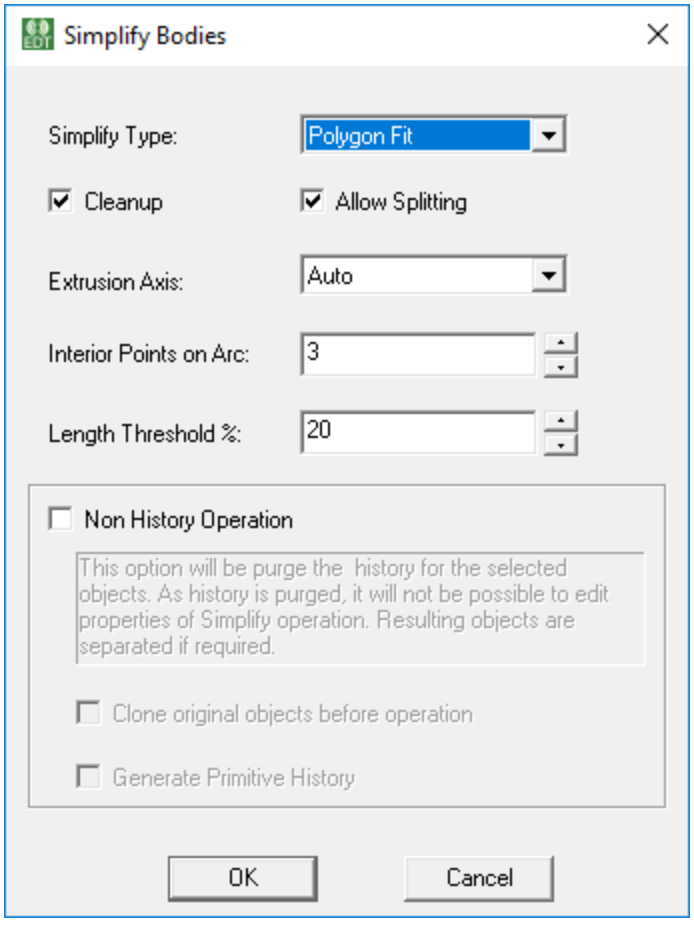


For Primitive Fit, the parameters are:

- Whether to Cleanup - This option allows you to clean the model before performing simplification. Cleanup include removing small features like rounds, fillets, chamfers in a solid body. It is recommended to set this option to get more simplified result.
- Whether to Allow Splitting - This option controls if object should be split during during simplification. If this option is selected, complex object will first split into multiple pieces and each piece will be further simplified. Setting this option will result in an object which will match more closely with original object. It is recommended to set this option.
- For the **Non History Operation** options, see below.

Polygon Fit Simplification

For Polygon Fit simplification, the object is replaced by set of polygon swept along normal and other primitives like box and cylinder. This simplification type typically produces geometry closest to original object with highest number of primitives. If you select Polygon Fit as the Simplify Type, the parameters are Clean Up, Allow Splitting, Extrusion Axis, Interior Points on Arc, and Length Threshold. For the **Non History Operation** options, see below.

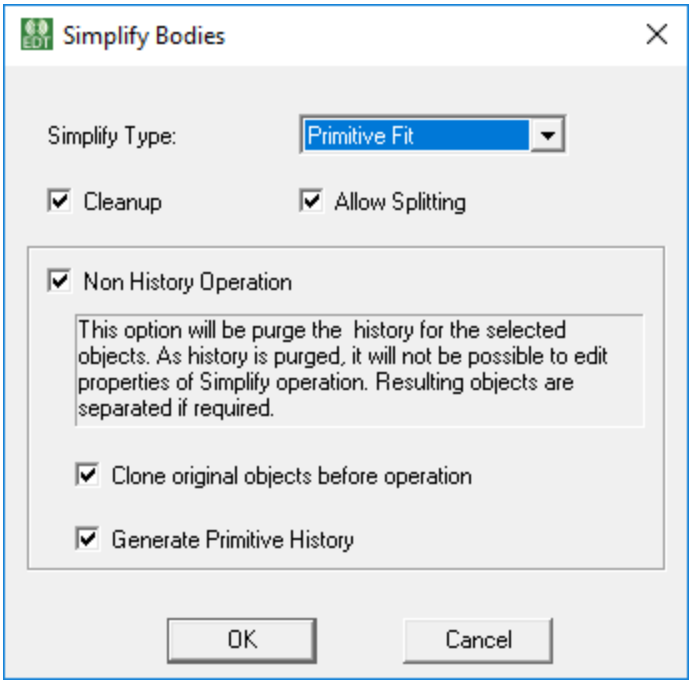


For Polygon Fit, you specify parameters for

- Whether to Cleanup - This option allows you to clean the model before performing simplification. Cleanup include removing small features like rounds, fillets, chamfers in a solid body. It is recommended to set this option to get more simplified result.
- Whether to Allow Splitting - This option controls if object should be split during during simplification. If this option is selected, complex object will first split into multiple pieces and each piece will be further simplified. Setting this option will result in an object which will match more closely with original object. It is recommended to set this option.
- Extrusion Axis, whether Auto selected, or X, Y, or Z. - This option specified a normal plane in which polygon profile is looked for. The default is Auto which allows an algorithm to determine best possible plane for polygon profile. You can specify a axis to help the algorithm, particularly when there is a draft or chamfer in the extrusion direction. The polygon profile is then swept in the extrusion direction.
- Interior Points on Arc - This specifies number of interior points created when a curve on the polygon profile is represented by line segments. The range is 1 to 100 with default being 3. Total number of points used to represent a curve is 5 when number of interior points is 3. This number specifies the maximum number of interior points to add. The number of points added could be less if the Length Threshold is already met.
- Length Threshold % - This specifies the length of edge as percentage of maximum length of an edge in the profile. It is specified as percentage of maximum edge length. If length of edge is more than the specified threshold, more interior points are added until the number of “Interior Points on Arc” is met.
- For the **Non History Operation** options, see below.

Non History Operation Options

The Non History Operation options are available for all Simplify Types.



With this option selected, the input body is simplified into multiple primitives (or polygon shapes) and a new part is created for each simplified shape. This option also purges the history of original part and it is not possible Edit Properties of the Simplify operation. You can choose to clone original object before simplification to retain history of original part. You can also choose to generate a primitive history.

Simplify result with separate bodies option

Use of Coordinate System

By default, the current working CS is used as the Simplify operation CS, but you can also change operation CS through the **Simplify** command property window or Property dialog.

Related Topics

Healing

Heal

The **Heal** command provides a way to correct geometric violations and to remove specific kinds of small features. When models are imported, two types of errors can occur – geometry errors and topology errors. Geometry errors are errors in definition of the underlying geometry while topology errors are errors in how the underlying components like faces, edges and vertices are connected. Ansys recommends that these be fixed before you invoke mesh generation.

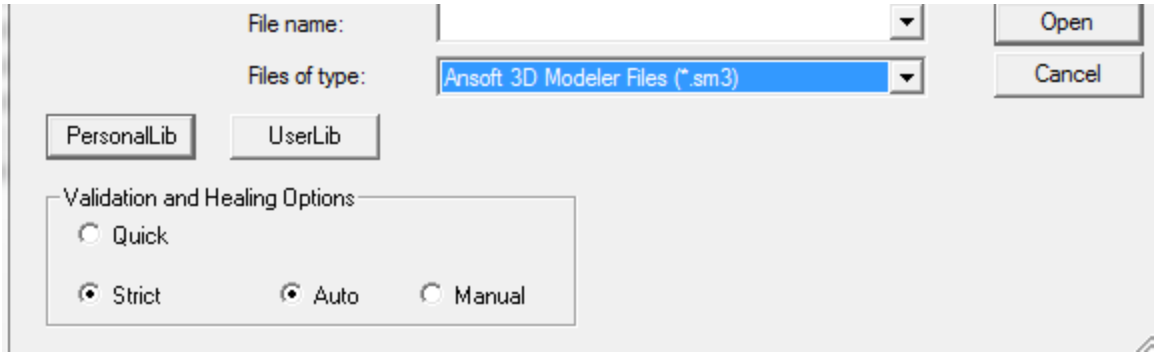
Imported objects which have only one operation on the history tree, can be healed. (Use the Purge History command to remove unwanted history operations before using **Heal**.)

Note	If you need to save the object history, save a separate copy for that purpose before you heal the object.

Healing can be invoked in different ways.

- The menu command **Modeler>Model Preparation>Heal** command applies to a selected object.
- Some formats permit healing during **Modeler>Import**. These are: 3D Modeler file (*.sm3), SAT file (*.sat), STEP file (*.step,*. stp), IGES file (*.iges, *.igs), ProE files (*.prt, *.asm), CATIA (*.model, *.CATpart), and Parasolid file (*.x_t, and *.x_b).

Selecting any of these formats enables a check box at the bottom of this window, "**Validation and Healing Options**."



See this table for details. The **Quick** option allows you to switch off healing to speed up the import process.

- The **Model Analysis** dialog that appears after running **Modeler>Model Analysis>Analyze Objects**, or **Modeler>Model Analysis>Show Analysis** dialog includes a **Perform** action menu with **Heal Objects** as a selection.

Any of these approaches leads to the same heal process.

Basic Steps in the Heal Process

There are several steps that are performed on selected objects.

1. Entity check, according to the **Analysis Options** settings.
2. Basic healing. This is done for all selected objects. Basic healing consists of fixing surface normals in the object and updating the orientation (to avoid having an object with negative volume).
3. Advanced healing. This is auto-heal. This is invoked on objects that require healing, that is, bodies that have errors, including have non-manifold errors.
4. Feature Removal. If you choose in the **Healing Options** to remove small holes, chamfers, blends, small edges, small faces and/or sliver faces, the actions are performed on all selected objects. There is no guarantee that small feature removal will be successful. (Also see Modifying the Model Resolution for defeaturing through the Auto Simplify and Model Resolution settings there.)

The above actions are performed on the selected objects. If you choose objects for healing which have not been analyzed, analysis is performed to determine its state (that is, whether it has invalid entities, small entities, and so forth). Invalid objects have all the above steps performed. Advanced healing is not performed on objects that do not require it.

While working on analyzing complex bodies, it is sometimes useful to examine faces, edges and vertices. In particular it is useful to find the connected faces for a face or edge or vertex, connected edges for a face/edge/vertex and connected vertices for a face/edge/vertex. The additional selection modes are available under **Edit->Selection Mode** and **Edit->Extend Selection**.

Related Topics

Align Faces
Remove Faces
Remove Edges
Healing Non-manifold Objects
Healing Options
Modifying the Model Resolution
Technical Notes: Removing Object Intersections
Technical Notes: Healing and Meshing
Technical Notes: Detecting and Addressing Model Problems to Improve Meshing

Validation and Healing Options for Import File

The Import File dialog contains shows different **Validation and Healing Options** for the seven file types listed here. The **Quick** option is allows you to import these formats without healing.

File Type	Validation and Healing Options Available			
3D Modeler file (*.sm3),	Quick	Strict		
SAT file (*.sat),	Quick	Strict	Auto or manual	
STEP file (*.step,*.stp),	Quick	Strict	Auto or Manual	Stitch Tolerance
IGES file (*.iges, *.igs)	Quick	Strict	Auto or Manual	Stitch Tolerance
ProE files (*.prt, *.asm)	Quick	Strict	Auto or Manual	Import Free Surfaces. This imports such surfaces as well as parts.
CATIA (*.model, *.CATpart)	Quick	Strict	Auto or Manual	Stitch Tolerance
Parasolid file (*.x_t, and *.x_b)	Quick	Strict	Auto or Manual	

Related Topics

Healing
Healing Options
Technical Notes: Removing Object Intersections
Technical Notes: Healing and Meshing

Technical Notes: Detecting and Addressing Model Problems to Improve Meshing

Technical Notes: *Mixed Dimensionality*.

Healing Stage One: Validation Check

After import an object, you should perform a validation check. This lets you focus on objects and object pairs that prevent the mesh from being invoked. The objects that fail api_check_entity() should be analyzed via the **Analyze Objects** menu item.

1. Select the objects, and click **Modeler >Model Analysis >Analyze Objects**.
   The **Analysis Options** dialog box appears, allowing you to perform small feature detections. Once you have completed this, click **OK**. The **Model Analysis** dialog box appears.
2. In the **Model Analysis** dialog box, select the objects marked with "Invalid Entities Found", and click **Perform>Heal Objects**. The **Healing Options** dialog box appears, allowing you to remove parameters. Click **OK**. The **Model Analysis** dialog box reappears.

Note	In most cases, the objects are healed, and the errors are fixed.

3. If errors persist, select the edges and faces still containing errors, and click **Delete**.
   This replaces each selected face/edge object by a tolerant edge/vertex, respectively. In some cases, the replacement of the face/edge by tolerant edge/vertex fails.

When models pass the initial validity checks, mesh generation could still fail. The following errors can be present in models: (See Detecting Errors.)

1. Non-manifold topology. These are non-manifold edges and vertices that are present in the model.
2. Object pair intersection. This detects whether pairs of objects intersect.
3. Small feature detection – small edge length, small face area and sliver face detection.
4. Mis-aligned entities detection – detects pairs of faces from objects that can be aligned to remove object intersections. This improves the probability of mesh success.
5. Mesh failure error display. This is available for single object, object pairs and last simulation run (all objects in a model). Errors reported by the meshing module are reported to the user.

Errors of type 3 and 4 must be resolved before the mesh can be invoked on the model.

By default, the **Heal** command is automatically applied to imported objects.

Related Topics

Healing

Fix Intersections Between Objects (Healing Stage 2)

Fix Object Pair Intersections (Healing Stage 3)

Analysis Phase (Healing Stage 4)

Healing Non-manifold Objects

Healing Options

Set Material Override

Technical Notes*: Healing and Meshing*

Technical Notes: *Detecting and Addressing Model Problems to Improve Meshing*

Healing Stage Two: Fix Intersections Between Objects

The second stage in healing is to fix intersections between objects. After validation check is performed, the pairs of objects that intersect are chosen for analysis. Use the analysis results to find whether objects have faces that can be aligned.

Choose all the bodies that intersect with another body.

1. From the **Objects** tab of the **Model Analysis** dialog box, click **Perform>Analyze Interobject Misalignment**.
   - Alternatively, you can click **Modeler>Model Analysis>Analyze Interobject Misalignment**.

   If the analysis finds object pairs that can be aligned, they are displayed on the **Objects Misalignment** tab.
2. You can select individual or multiple rows, and align the faces. In some cases, face alignment fails if the topology of the body changes by a large factor after alignment.
3. Identify individual bodies and body pairs that fail to mesh.
4. Perform a mesh analysis on individual objects and object pairs.
5. Review the reports and fix the errors.

Related Topics

Healing

Validation Check (Healing Stage 1)

Fix Object Pair Intersections (Healing Stage 3)

Analysis Phase (Healing Stage 4)

Healing Non-manifold Objects

Healing Options

Set Material Override

Technical Notes*: Healing and Meshing*

Technical Notes: *Detecting and Addressing Model Problems to Improve Meshing*

Healing Stage Three: Fix Object Pair Intersections

The next stage of healing is to fix object-pair intersections. The healing process in stage two might fix some of these errors (by alignment).

1. If these errors are still present, you must remove them either by using the **Remove Faces** command (**Modeler>Model Analysis>Remove Faces**) or by performing a Boolean subtraction.
2. Overlap between objects is too large to be fixed by healing or by face alignment. Boolean intersect shows the common portion between the bodies. In this case, use a subtract operation to remove overlaps.

Related Topics

Healing
Validation Check (Healing Stage 1)
Fix Intersections Between Objects (Healing Stage 2)
Analysis Phase (Healing Stage 4)
Healing Non-manifold Objects
Healing Options
Set Material Override
Technical Notes*: Healing and Meshing*
Technical Notes: *Detecting and Addressing Model Problems to Improve Meshing*

Healing Stage Four: Analysis Phase

If the last simulation run contains errors, they are displayed in the **Model Analysis** dialog box, on the **Mesh Feedback** tab.

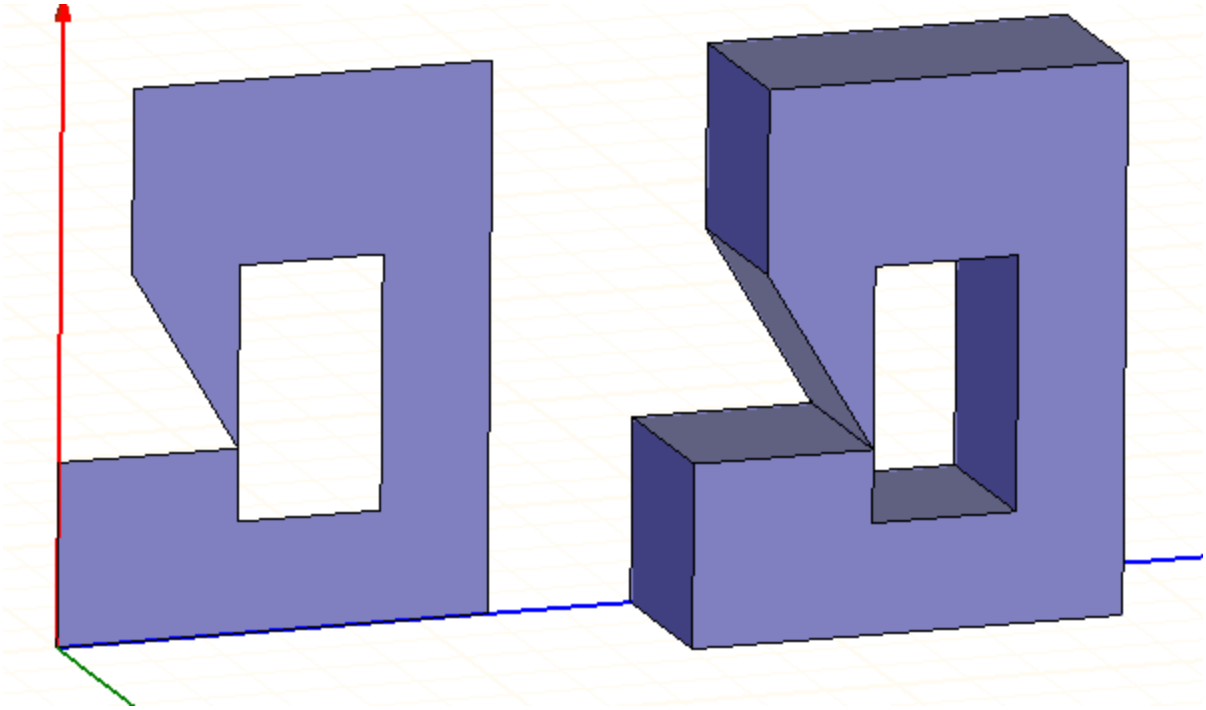
Related Topics

Healing
Validation Check (Healing Stage 1)
Fix Intersections Between Objects (Healing Stage 2)
Fix Object Pair Intersections (Healing Stage 3)
Healing Non-manifold Objects
Healing Options
Set Material Override
Technical Notes*: Healing and Meshing*
Technical Notes: *Detecting and Addressing Model Problems to Improve Meshing*

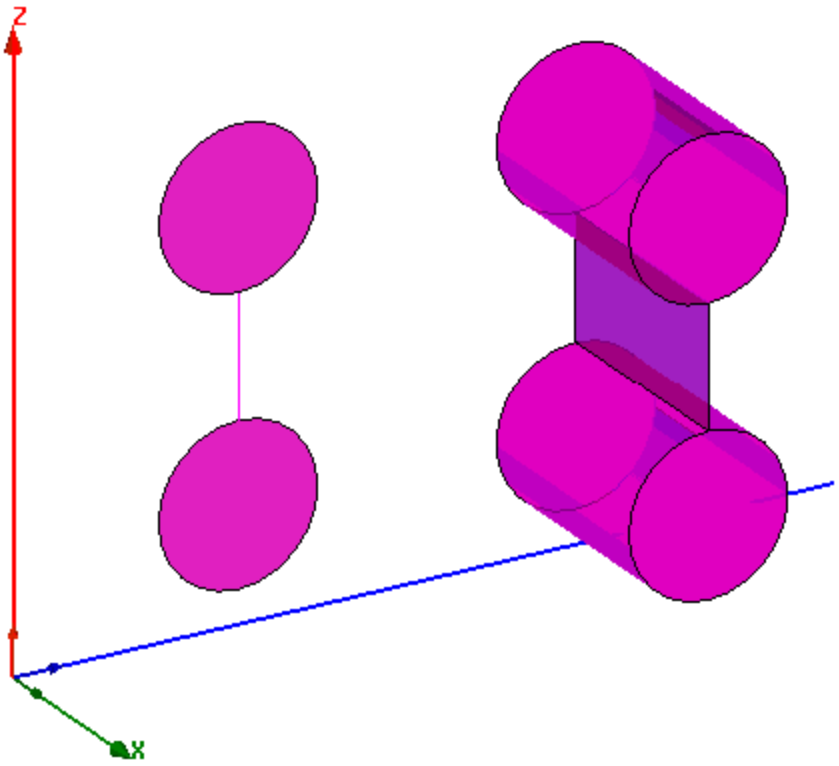
Healing Non-Manifold Objects

Non-manifold objects, in simple terms, are non-physical objects or objects that cannot be manufactured. For example, objects that intersect themselves (like the symbol for infinity in 2d)

are clearly non-manifold. In addition objects that touch themselves may be non-manifold such as when a 2D object touches itself at a vertex, or a 3D object touches itself at a point or edge. These cases are shown in the following figure.



Another type of non-manifold object has mixed dimensionality. For example, a pair of 2D objects connected by a 1D line segment, or a pair of 3D objects connected by a 2D sheet object. These cases are illustrated below.



The criteria for manufacturability is a simple manifestation of a complex mathematical concept that must be adhered to in the solid modeling system. When creating geometry, either directly, or through boolean operations, you should always consider whether or not the resulting operation will result in an object that could not be manufactured. If this is the case, then the object will cause an error in the modeler or in the meshing system.

To heal non-manifold objects:

1. Identify an edge that is non-manifold.
2. Select the connected faces.

   You can use Face selection.
3. Create a face coordinate system on the planar face.
4. Create a small box to cover the non-manifold edge.
5. Either do a union or a subtraction to remove the faces that contain the non-manifold edge.

   The non-manifold edge is now removed. You may also remove or add a small portion of the model.
6. Repeat the above steps for all the non-manifold edges.

Related Topics

Healing

Validation Check (Healing Stage 1)

Fix Intersections Between Objects (Healing Stage 2)
Fix Object Pair Intersections (Healing Stage 3)
Analysis Phase (Healing Stage 4)
Healing Options
Set Material Override
Technical Notes:*Healing and Meshing*
Technical Notes: *Detecting and Addressing Model Problems to Improve Meshing*

Related Topics

Healing
Technical Notes: *Removing Object Intersections*
Healing Options
Technical Notes*: Healing and Meshing*
Technical Notes: *Detecting and Addressing Model Problems to Improve Meshing*
Technical Notes: *Mixed Dimensionality*

Setting the Healing Options

The **Healing Options** let you control how healing proceeds with respect to a variety of features and issues.

1. Click **Modeler>Model Healing>Heal** to open the **Healing Options** dialog box. You can also open the **Healing Options** dialog from the **Model Analysis** dialog via the **Objects** tab drop down menu.

   The **Healing Options** dialog contains three tabs:
   - **Healing Options**
   - **Feature Removal Options**
   - **Properties**, which lists the geometric properties of the currently selected object.
2. Select the **Healing Options** tab on the **Healing Options** dialog to specify the following:
   - **Heal Type** as: **Auto Heal** (default), **Manual Heal**, or **No Heal**.

   Selecting **Manual Heal** enables the **Manual Heal Options**:
   - **Perform Tolerant Stitching** check box.

   This enables a field for the **Stitch Tolerance** value, and a check box to **Stop After First Error**.

   - **Perform Geometry Simplification**

   This enables fields for **Simplification Tolerance** and **Maximum Generated Radius** values.

   You can also select radio buttons to **Simplify Curves**, **Surfaces**, or **Both**.

- **Tighten Gaps** settings.

  A check box to select **Perform Tighten Gaps**

  A field to specify **Tighten Gaps Within** a given value in mm.

3. Select the **Feature Removal Options** tab to specify the following:

   Here you can specify the following **Feature Removal Options**.
   - **Remove Holes** check box and **Maximum Radius** value.
   - **Remove Chamfers** check box and **Maximum Width** value.
   - **Remove Blends** check box and **Maximum Radius** value.

   You can specify the following **Remove Small Entity Options**:
   - **Small Edges, Length Less Than**, less than a specified value.
   - **Small Faces Area e Less Than**, less than a specified area.
   - **Sliver Face Width Less Than**, less than either:
     - **Object Bounding box Scale Factor**, less than a specified scale factor
     - **Sliver Edge Width**, less than a specified value.

Note	Sliver faces have a maximum distance among the long edges that is smaller than the specified tolerance and have at least one short edge and at most three long edges. A short edge has a length less than the specified tolerance. A long edge has a length greater than the specified tolerance. You can give the tolerance as a absolute value or a factor of the bounding box containing the face.

   You can **Control Object Properties Change** according to the following settings:
   - **Allowable Change in Surface Area** check box, and percent value.
   - **Allowable Change in Volume** check box, and percent value.
4. Select the **Properties** tab to view the geometric properties of the currently selected object.
5. Click **OK** to apply the specified Healing options and to open the Analysis dialog.

Related Topics

Healing
Validation Check (Healing Stage 1)
Fix Intersections Between Objects (Healing Stage 2)
Fix Object Pair Intersections (Healing Stage 3)
Analysis Phase (Healing Stage 4)
Healing Non-manifold Objects
Set Material Override
Technical Notes*: Healing and Meshing*
Technical Notes: *Detecting and Addressing Model Problems to Improve Meshing*

Related Topics

Healing
Stitch Sheets
Technical Notes: *Removing Object Intersections*
Healing Non-manifold Objects
Modifying the Model Resolution
Technical Notes*: Healing and Meshing*
Technical Notes: *Detecting and Addressing Model Problems to Improve Meshing*

Stitch Sheets

Use the **Modeler>Model Healing>Stitch Sheets** command to stitch selected sheets.

1. Select two or more sheet objects.

   This enables the Stitch Sheets command on the **Modeler>Model Healing** submenu.
2. Click **Modeler>Model Healing>Stitch Sheets**

   This displays a Stitch dialog with a Maximum Stitch Tolerance field. The default value (auto) comes from the **Healing** dialog **Options** tab with Manual Healing selected. You may edit the value in the Stitch dialog or in the Healing Options.
3. Click OK.

   This closes the dialog and attempts to perform stitching on the selected sheets. If the sheets are separated beyond the stitch tolerance, stitching is not performed and a warning is issued.

Related Topics

Healing

Removing Selected Faces

Use the **Modeler>Model Preparation>Remove Faces** command to remove the selected faces. You can also use the Model ribbon **Remove Faces** icon when you have made an appropriate face selection.

If you find object-pair intersections that healing does not fix, or that can be fixed (by alignment), you can correct the problem by one of the following methods.

1. Use the **Remove Faces** command (**Modeler>Model Healing>Remove Faces**) or by performing Boolean subtract.
2. If overlap between objects is too large to be fixed by healing or by face alignment. Boolean intersect shows the common portion between the bodies. In this case, use a subtract operation to remove overlaps.

Related Topics

Align Faces
Analyzing Objects
Analyzing Interobject Misalignment
Analyzing Surface Mesh
Healing an Imported Object
Healing
Technical Notes: Healing and Meshing
Technical Notes: Healing Models
Technical Notes: Error Types
Technical Notes: Detecting Errors

Removing Selected Edges

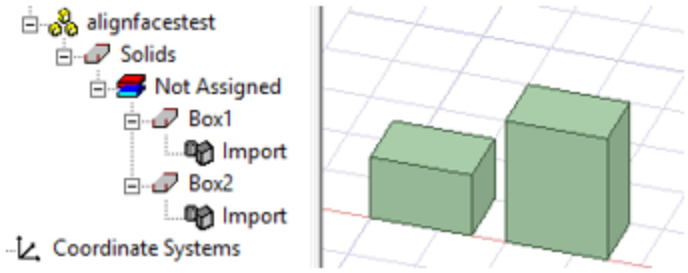
Use this **Modeler>Model Healing>Remove Edges** command to remove the selected edges.

Related Topics

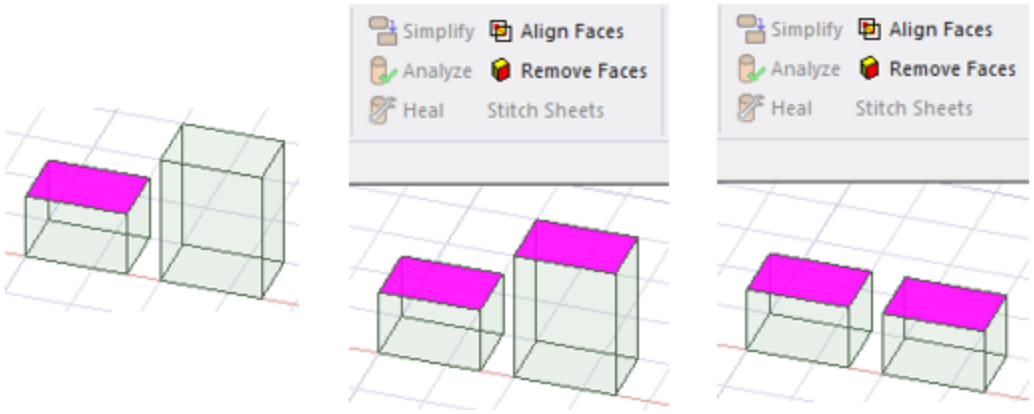
Analyzing Objects
Analyzing Interobject Misalignment
Analyzing Surface Mesh
Healing an Imported Object
Technical Notes: Healing and Meshing
Technical Notes: Healing Models
Technical Notes: Error Types
Technical Notes: Detecting Errors

Aligning Selected Faces

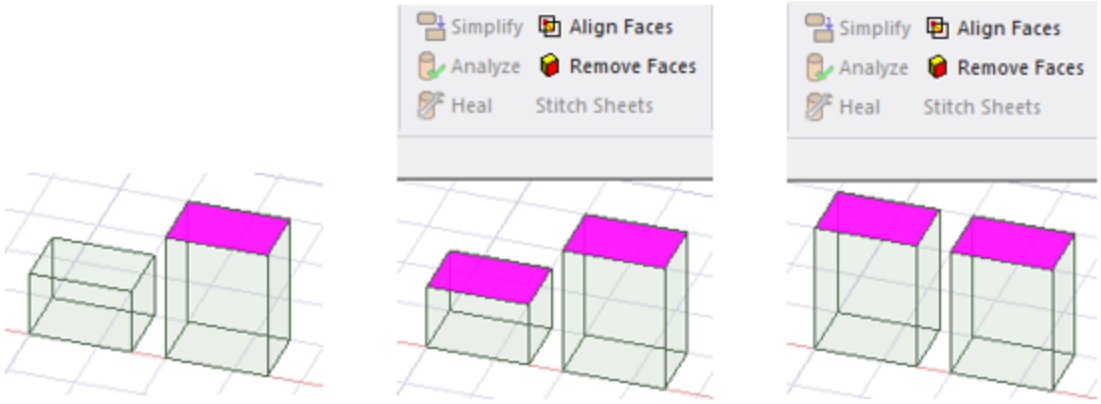
Use the **Modeler>Model Preparation>Align Faces** command to align the adjacent selected faces of imported objects which have only one operation in their **History** tree. This command is **Undo**-able.



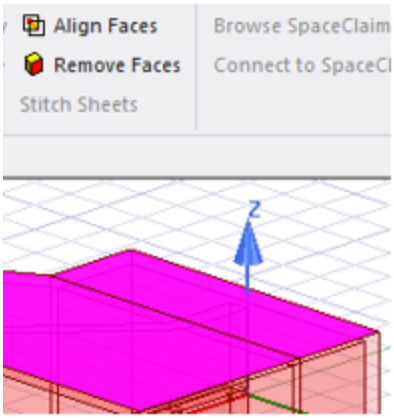
The **Model** ribbon also includes an **Align Faces** icon, which is enabled when you make appropriate face selection. The first selected face defines the plane of alignment for subsequent selection, that is, tool faces to blank faces.

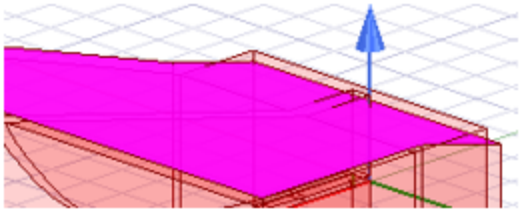


Selecting faces in a different order changes the alignment.



The selected faces can touch.





Related Topics

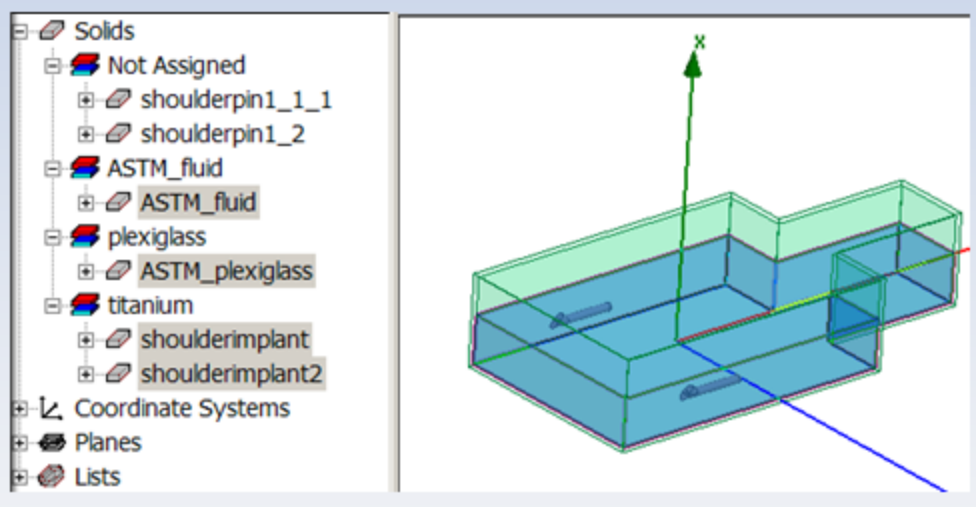
Analyzing Objects
Analyzing Interobject Misalignment
Analyzing Surface Mesh
Healing

Performing Explicit Subtractions

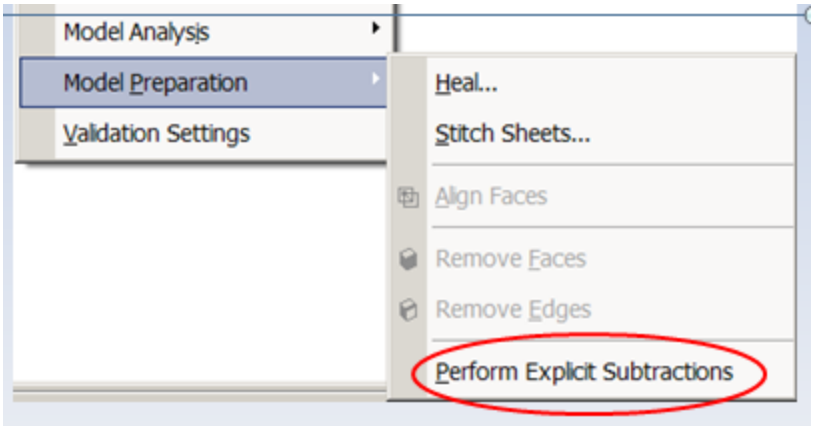
Use the **Modeler>Model Preparation>Perform Explicit Subtractions** command to subtract selected **History** tree objects that are fully contained by other parts.

1. Select two or more history objects.

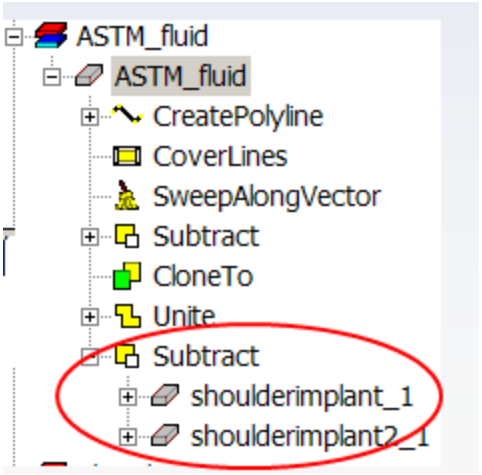
   This enables the Stitch Sheets command on the **Modeler>Model Preparation>Perform Explicit Subtractions** submenu. For example, consider the following model.



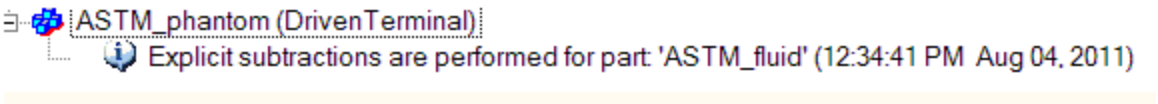
2. **Perform Explicit Subtraction** can be performed.



The results appear in the History tree as shown:



The Message window also reports this action.



If no parts are fully contained by other parts, a message explains that no explicit subtraction is needed.

Related Topics

Healing

Modifying Objects

You can quickly modify the position, dimensions, and other characteristics of objects created in the **Modeler** window.

What do you want to do?

- Assign color to an object
- Assign transparency to an object
- Assigning a Cross Section and Dimension to a Polyline
- Copy and paste objects
- Delete objects
- Delete Last Operation
- Cutting Objects
- Move objects
- Rotate objects
- Change the Orientation of an object

- Mirror objects about a plane
- Offset an object (move every face of an object)
- Duplicate objects
- Scale the size of objects
- Sweeping objects
- Cover lines
- Cover faces
- Uncover faces
- Detach faces
- Detach edges
- Create a new object by taking a cross-section of a 3D object
- Connect objects
- Move faces
- Move Edges
- Unite objects
- Subtract objects
- Create objects from intersections
- Create an object from a face
- Create an object from an edge
- Split objects
- Separate objects
- Convert polyline segments.
- Round the edge of an object (Fillet)
- Flatten the edge of an object (Chamfer)
- Wrap Sheet Command
- Imprint Projection Commands
- Imprinting an Object
- View and Edit Commands on History Tree Objects
- Purge History
- Generate History

Assigning Color to Objects

1. Select the object to which you want to assign a color.

Note	If the **Properties** window not visible on the desktop, click **View>Properties Window**.

2. In the **Properties** window on the desktop, click the **Attribute** tab.
3. Click **Edit** in the **Color** row.

   The **Color** palette appears.
4. Select a color from the **Color** palette, and then click **OK**.

The color is assigned to the selected object.

5. Click OK to close the **Properties** dialog box.

Related Topics

Setting the Default Color of Objects

Setting the Default Color of Objects

1. Click **Tools>Options>Modeler Options**.
2. Click the **Display** tab.
3. Select **Object** from the **Default color** pull-down list.
4. Click the color button beside the **Default color** pull-down list.
   The **Color** palette appears.
5. Select a color from the **Color** palette, and then click **OK**.
   Any objects you draw after this point are assigned the default color you selected.

Setting the Default Color of Object Outlines

1. Click **Tools>Options>Modeler Options**.
2. Click the **Display** tab.
3. Select **Object Wire** from the **Default color** pull-down list.
4. Click the color button beside the **Default color** pull-down list.
   The **Color** palette appears.
5. Select a color from the **Color** palette, and then click **OK**.
   The outlines of any objects you draw after this point are assigned the default color you selected.

Assigning Transparency to an Object

1. Select the object to which you want to assign a transparency.

Note	If the **Properties** window not visible on the desktop, click **View>Properties Window** or use **Edit>Properties**.

2. In the **Properties** dialog box, click the **Attribute** tab.
3. Click the value in the **Transparency** row.
   The **Set Transparency** window appears.
4. Move the slider to the right to increase the transparency of the object. Move the slider to the left to decrease the transparency of the object.
5. Click **OK**.

Related Topics

Setting the Default Transparency of Objects

Setting the Default Transparency of Objects

1. Click **Tools**>**Options**>**Modeler Options**.
2. Click the **Display** tab.
3. Move the **Default transparency** slider to the right to increase the transparency of objects.

   Move the slider to the left to decrease the transparency of objects.

   Any objects you draw after this point are assigned the default transparency you selected.

Copying and Pasting Objects

To copy objects and paste them in the same design or another design, use the **Edit>Copy** and **Edit>Paste** commands. For data link purposes, where you want to include the material assignments as well as the geometry, you can use the Import from Clipboard command.

1. Select the objects you want to copy.
2. Click **Edit**>**Copy**  .

   The objects are copied to the Clipboard, a temporary storage area. The selected items are not deleted.
3. Select the design into which you want to paste the objects. It can be the same design from which you copied the items.
4. Click in the **Modeler** window.
5. Select the working coordinate system. Objects are pasted relative to the current working coordinate system.
6. Click **Edit**>**Paste**  .

The objects appear in the targeted Modeler window and in the new History tree. The pasted objects become the current active selection. If you undo the paste, the active selection goes back to the previous selection. If you redo, the pasted objects become the current active selection.

Items on the Clipboard can be pasted repeatedly. The items currently stored on the Clipboard are replaced by the next items that are cut or copied.

Related Topics

Duplicating Boundaries and Excitations with Geometry

Copying an Image to the Clipboard

Use this command to copy a 3D view of the design, field plots, or 2D reports to the Clipboard and paste it into another document, such as Word or PowerPoint.

To copy an image to the Clipboard:

1. Make the window of the image you wish to copy active.

   This enables the **Edit>Copy Image** command in the menu bar.
2. Click **Edit>Copy Image**, or right click on the active window to display the shortcut menu and select **Copy Image**.

   The report is copied to the Clipboard as an image.
3. Select and open the application into which you want to paste the objects, and paste the image.

Deleting Objects

1. Select the objects to delete.
2. Click **Edit**>**Delete**  .
   - Alternatively, press **Delete**.

   The objects are deleted.

Note	To maintain valid boundaries, excitations, or other parameters that were associated with the deleted object, reassign them to other objects.

Related Topics

Deleting Start Points and Endpoints

Deleting Start Points and Endpoints

If you select a polyline in the history tree, the **Delete Start Point** and **Delete End Point** commands may be enabled. These permit you to delete portions of the line.

1. In the history tree, locate the polyline that contains the segment you want to delete. Expand this part of the history tree.
2. In the history tree, select the polyline you want to edit.

   The segment is highlighted.
3. On the **Edit** menu or the shortcut menu, click either **Delete Start Point to remove the leading segments or Delete End Point to remove the following segments.**

   The designated segment is removed, and the line changes.

Deleting Last Operation

This undoes the last operation, including removing that operation from the history, and updating the context for the **Undo** and **Redo** commands.

Related Topics

Undoing Commands
Redoing Commands

Cutting Objects

1. Select the objects to cut.
2. Click **Edit**>**Cut**.

The objects are copied to the Clipboard and deleted from the design.

Moving Objects

1. Select the objects to move.
2. Click **Edit**>**Arrange**>**Move**  .
3. Select an arbitrary anchor point in one of the following ways:
   - Click the point.
   - Enter the point's coordinates in the **X**,**Y**, and **Z** boxes.
4. Select a target point in one of the following ways:
   - Click the point.
   - Type the coordinates of a point relative to the anchor point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   All selected objects move the distance determined by the offset between the anchor point and the target point.

Rotating Objects

Rotate objects about the x-, y-, or z-axis using the **Edit>Arrange>Rotate** command.

To rotate objects about an axis:

1. Select the objects to rotate.
2. Click **Edit**>**Arrange**>**Rotate**  .

   The **Rotate** dialog box appears.

3. Select the axis about which to rotate the objects: **X**, **Y**, or **Z**.
4. Type the angle to rotate the objects in the **Angle** box.

   A positive angle causes the object to be rotated in the counter-clockwise direction. A negative angle causes the object to be rotated in the clockwise direction.
5. Click **OK**.

   The selected objects are rotated about the axis.

To rotate *andcopy* objects, use the **Edit>Duplicate>Around Axis** command.

Changing the Orientation of an Object

Each object has an **Orientation** property that specifies the coordinate system it uses is Global, or a user defined orientation relative to the Global coordinate.

This property is useful in dealing with anisotropic materials. The properties of anisotropic materials are specified relative to the objects orientation. Changing the orientation of an object provides a way for objects made of the same material to be orientated differently.

To change an object's orientation.

1. Define the coordinate systems you want to have available.
2. Open the properties window for the object.
3. Click on the Orientation property, and select from the Drop down list. If no Orientations other than Global have been defined, none appear on the list.
4. Click **OK** to close the dialog and apply the changes.

Related Topics

Assigning Material Property Types

Setting Coordinate Systems

Creating a Relative Coordinate System

Mirroring Objects

Mirror an object about a plane using the **Edit>Arrange>Mirror** command. The plane is selected by defining a point on the plane and a normal point. This command allows you to move an object and change its orientation.

Note	The distance between the point on the mirror plane and the point along the normal does not matter — only the vector direction matters.

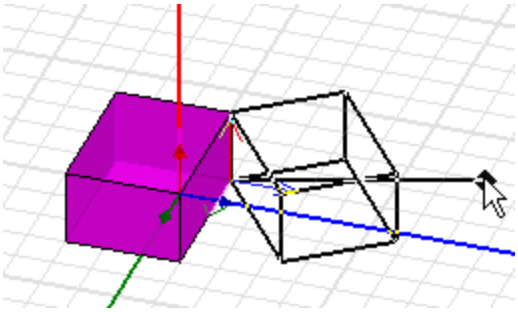
To mirror an object about a plane:

1. Select the object you want to mirror. You can select multiple objects.

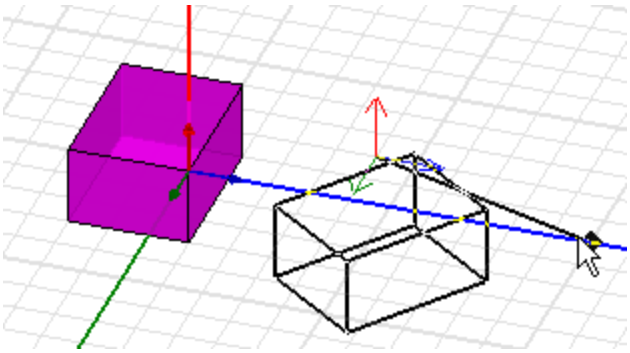
1. Click **Edit**>**Arrange**>**Mirror**  .
2. Select a point on the plane on which you want to mirror the object.

You can do this by clicking a point, or typing coordinates in the X, Y, and Z boxes in the status bar.

If you select a point on the object, the mirroring is relative to that point on the object. In the following example, the first point clicked after selecting **Edit>Arrange>Mirror** was on the right-rear bottom corner of the selected object. So the axis of rotation as you move the cursor is that corner. As you move the cursor, it drags a diamond-shape on a vector extending from the initial point. The distance along the vector does not matter. Moving the mouse rotates an outline of the object to new orientations. Clicking the mouse moves the object to location indicated by the outline.



In this second example, the initial point is at a distance from the original object, designated by the triad from which the handle for rotation extends to the dragging cursor.



3. Select a normal point in one of the following ways:
   - Click the point.
   - Type the coordinates of a point relative to the first point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point. As you type in the values, the outline moves to the coordinates. Press the Enter key to complete the command.

Note	For 2D designs, movement is restricted to the XY or RZ plane.

   The selected object is moved to the plane you specified and oriented relative to the normal point you specify.

To mirror *and copy* objects about a plane, use the **Edit>Duplicate>Mirror** command.

Related Topics

Duplicating and Mirroring Objects

Offsetting Objects

Move every face of a 3D object in a direction normal to its surface using the **Edit>Arrange>Offset** command. The faces are moved a specified distance normal to their original planes. This command enables you to move every face of a solid object without having to individually select and move each face. Use the **Surfaces>Move Faces>Along Normal** command if you want to move just one or more faces of an object.

To offset every face of an object:

1. Select the object you want to offset.
2. Click **Edit>Arrange>Offset**.

   The **Offset** dialog box appears.
3. Type the distance you want to move the object faces from their origins, and then select a unit from the pull-down list.
4. Click **OK**.

   The selected object's faces are moved the distance you specified.

Duplicating Objects

You can duplicate objects within a design using the **Edit>Duplicate** commands. Duplicates are dependent upon the parameters of their *parent* object at the time they were created, that is, they share the parent object's history at the time of creation. The command hierarchy in the history tree shows the duplication command, illustrating which commands affect all duplicates (those performed before the duplication) and which commands do not affect the duplicates (those performed after the duplication). For example, if you modify the radius of a parent object's hole, the change is applied to the holes of the object's duplicates because they share the radius specification history, but if you move the faces of the parent object, its duplicates are not affected because this operation took place after the duplicates were created.

Operations performed on duplicates are independent. For example, if you duplicate a cylinder twice, creating a row of three, and then split the second cylinder, the first and third cylinders are not affected by the split.

When creating duplicates, the parent object is duplicated along a line or around an axis the number of times you specify. You can also create a single duplicate that mirrors the parent object about a plane.

Choose from the following commands:

Edit>Duplicate>Along Line	Duplicates the parent object along a straight line.

Edit>Duplicate>Around Axis	Duplicates the parent object around an axis.
Edit>Duplicate>Mirror	Duplicates a mirror image of the parent object about a plane.

To copy objects to another design, use the **Edit>Copy** and **Edit>Paste** commands.

Note	There is currently no method for dissolving the parent/duplicate relationship once a duplicate has been created.

Duplicating Objects Along a Line

You can duplicate an object along a straight line using the **Edit>Duplicate>Along Line** command. The line along which the object is duplicated can be vertical, horizontal, or lie at an angle.

1. Select the object you want to duplicate.
2. Click **Edit**>**Duplicate>Along Line**  ..
3. Specify the vector along which the object is to be duplicated:
   a. Select an arbitrary anchor point in one of the following ways:
      - Click the point.
      - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.

   Any point in the drawing region can be selected; however, selecting an anchor point on the object's edge or within the object makes it easier to select the duplication line.

   b. Select a second point in one of the following ways:
      - Click the point.
      - Type the coordinates of a point relative to the anchor point in the **dX**, **dY**, and **dZ** boxes.

   This point defines the direction and distance from the anchor point to duplicate the object.

   The **Duplicate along line** dialog box appears.
4. Type the total number of objects, including the original, in the **Total number** box.
5. By option check the **Attach to Original Object** check box. If this is checked, no ports or boundary conditions are duplicated for the child.
6. Click **OK**.

   The duplicates are placed along the vector you specified.

Duplicating Objects Around an Axis

To duplicate an object around the x-, y-, or z-axis, use the **Edit>Duplicate>Around Axis** command.

1. Select the object you want to duplicate.
2. Click **Edit>Duplicate>Around Axis**  .

   The **Duplicate Around Axis** dialog box appears.
3. Select the axis around which you want to duplicate the object: **X**, **Y**, or **Z**.
4. Type the angle between duplicates in the **Angle** box.

   A positive angle causes the object to be pasted in the counter-clockwise direction.

   A negative angle causes the object to be pasted in the clockwise direction.
5. Type the total number of objects, including the original, in the **Total number** box.
6. By option check the **Attach to Original Object** check box. If this is checked, no ports or boundary conditions are duplicated for the child.
7. Click **OK**.

   The object is duplicated around the axis at the angle you specified.

Duplicating and Mirroring Objects

To duplicate and mirror an object about a plane, use the **Edit>Duplicate>Mirror** command. The plane is selected by defining a point on the plane and a normal point. This command allows you to duplicate an object and specify the duplicate's position.

This command is similar to **Edit>Arrange>Mirror**, except that this command duplicates an object, rather than moves it.

1. Select the object you want to mirror.
2. Click **Edit>Duplicate>Mirror**  .
3. Select a point on the plane on which you want to mirror the object.

   A line drawn from this point to the mirror plane will be perpendicular to the plane. The distance between the point on mirror plane and point along the normal does not matter; only the vector direction matters
4. Select a normal point on the plane in one of the following ways:
   - Click the point.
   - Type the coordinates of a point relative to the first point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

Note	For 2D designs, movement is restricted to the XY or RZ plane.

   A duplicate of the object appears on the plane you specified, oriented according to the normal point you specified.

Related Topics

Mirroring Objects

Scaling Objects

Scale an object's dimensions in one or more directions using the **Edit>Scale** command.

The scale of an object is determined by the distance of each of its vertices from the origin of the model coordinate system. When an object is scaled, the distance of each vertex from the origin is multiplied by the scaling factor, causing the object to be resized and/or moved.

For example, if you specify a scaling factor of 2 in the x direction, each vertex in the model is moved so that the distance to its origin is doubled. A vertex located at the origin cannot not move. You can alter an object's proportions by scaling it in one direction.

To scale an object's dimensions in one or more directions:

1. If necessary, set a different working coordinate system to achieve the desired scaling.
2. Select the object to scale.
3. Click **Edit>Scale**.

   The **Scale** dialog box appears.
4. Type the scale factor for each axis.
5. Click **OK**.

   The object is scaled about the working coordinate system's origin.

Related Topics

Modifying Object Attributes using the Properties Window

Sweeping Objects

You can sweep a 2D object around an axis, along a vector, or along a path to create a 3D solid object. Objects that can be swept include circles, arcs, rectangles, trapezoids, polylines, or any 2D object created in the **3D Modeler** window. The 2D object need not be orthogonal to the sweep path.

You can also thicken sheets to make a 3D object.

You can also sweep open 1D objects, such as polylines. This results in open 2D sheet objects.

You can also sweep one or more faces of a 3D object to create a new object. See Sweep Faces Along Normal.

Related Topics

Assigning a Cross Section and Dimension to a Polyline
Sweeping Around an Axis
Sweeping Along a Vector
Sweep Along a Path
Sweep Faces Along Normal
Draft Types

Thicken Sheet

Sweeping Around an Axis

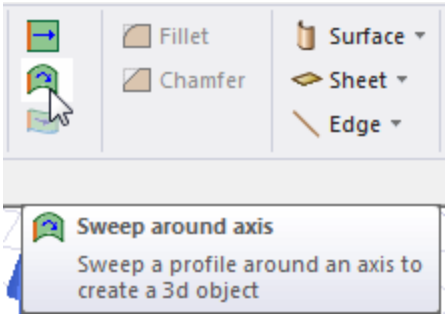
Sweep a 1D or 2D object around the x-, y-, or z-axis using the **Draw>Sweep>Around Axis** command. Sweeping circles around an axis is a convenient way to create an open coil loop.

Before using this command, keep the following guidelines in mind:

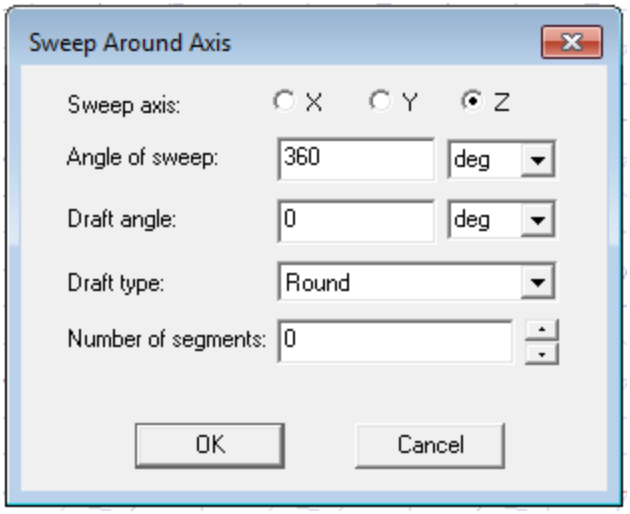
- The object and the axis you are sweeping around must lie in the same plane. For example, if you are sweeping an object around the z-axis, the object must lie in a plane that includes the z-axis, such as xz or yz.
- The normal of the object's plane faces must be perpendicular to the axis around which you are sweeping.
- The object may not cross the axis around which it is being swept.

To sweep an object around an axis:

1. Select the object you want to sweep.
2. From the menu bar, click **Draw> Sweep> Around Axis** or, in the **Draw** ribbon tab, click **Sweep around axis**:



The **Sweep Around Axis** dialog box appears.



3. Select **X**, **Y**, or **Z** as the **Sweep axis**.
4. Type an angle in the **Angle of sweep** text box, and select the unit of measurement. The value must be between **-360** and **360** degrees.
5. Type an angle into the **Draft angle** text box and select the desired unit of measure.

   This is the angle to which you want a 2D object's profile, or shape, expanded (positive angle) or contracted (negative angle) as it is swept. Among other purposes, drafts make it easy to remove molded or cast parts from their molds.

   For 1D objects, the draft angle progressively alters the sweep radius but does not alter the length of the object as it is swept. For example, a 10 mm straight line swept around a parallel axis with a nonzero draft angle would produce a spiral surface with a fixed width of 10 mm. When the sweep angle is *positive*, a positive draft angle produces a spiral surface with a *decreasing* radius (that is, the distance from the generated surface to the sweep axis decrease along the sweep angle. A negative angle produces a spiral surface with an *increasing* radius along the sweep angle. When the sweep angle is *negative*, the spiral behavior is the opposite of that just described.
6. Select one of the following **Draft type** options from the drop-down menu. The draft type instructs the modeler whether to maintain the original object shape and how to transition between adjacent faces. This setting affects the results when sweeping 2D objects into 3D solids or when sweeping lines and curves into 3D sheets.
   - **Extended**
   - **Round**
   - **Natural**

   For a complete description of the behavior of each of these three options, see the Draft Types page.
7. Type the number of segments in the **Number of segments** text box.

Note	The default number of segments is zero, which creates a true path. A positive value results in a segmented sweep, while a negative value results in an error. If the sweep angle is 360 degrees, the number of segments is equal to the value specified. If the sweep angle is less than 360 degrees, half segments appear at the ends. Projects and scripts from previous software versions are treated as if the number of segments were zero.

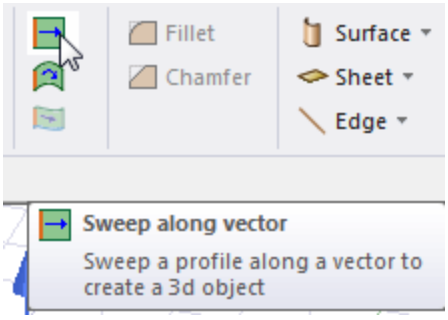
8. Click **OK**.

   The object is swept around the axis. The new object has the properties of the original object.

Sweeping Along a Vector

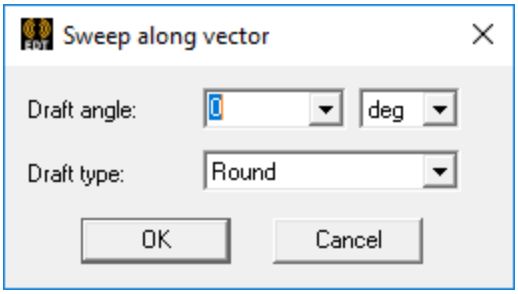
Sweep a 1D or 2D object along a vector using the **Draw>Sweep>Along Vector** command.

1. Select the object you want to sweep.
2. From the menu bar, click **Draw> Sweep> Along Vector** or, in the **Draw** ribbon tab, click **Sweep along vector**:



3. Draw the vector you want to sweep the object along:
    a. Select the start point by clicking the point or typing its coordinates in the **X**, **Y**, and **Z** boxes.
    b. Select the endpoint in one of the following ways:
        - Click the point.
        - Type the coordinates of a point relative to the start point in the **dX**, **dY**, and **dZ** boxes, where **d** is the relative distance from the previously selected point.

    The **Sweep along vector** dialog box appears.



4. Type an angle into the **Draft angle** text box and select the desired unit of measure.

    This is the angle to which you want a 2D object's profile, or shape, expanded (positive angle) or contracted (negative angle) as it is swept. Among other purposes, drafts make it easy to remove molded or cast parts from their molds.

    For 1D objects, the draft angle alters the sweep direction but does not alter the length of the object as it is swept. For example, a 10 mm straight line swept with a 5-degree draft angle does not produce a trapezoidal 2D object. It produces a rectangle with a constant 10 mm width but tilted 5 degrees relative to the specified sweep vector.

5. Select one of the following **Draft type** options from the drop-down menu. The draft type instructs the modeler whether to maintain the original object shape and how to transition between adjacent faces. This setting affects the results when sweeping 2D objects into 3D solids or when sweeping lines and curves into 3D sheets.
   - **Extended**
   - **Round**
   - **Natural**

   For a complete description of the behavior of each of these three options, see the Draft Types page.
6. Click **OK**.

   The object is swept along the vector. The new object has the name and color of the original profile. The **Properties** dialog box appears, enabling you to modify the object's properties.

Sweeping Along a Path

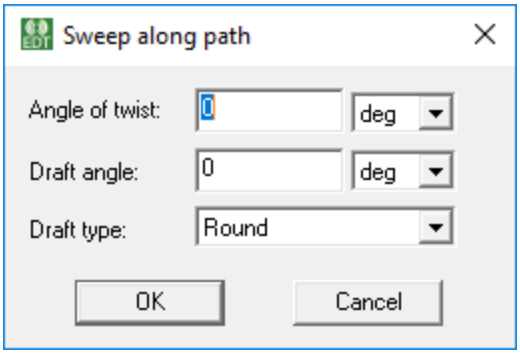
Sweep a 1D or 2D object along a path that is defined by an open or closed polyline using the **Draw>Sweep>Along Path** command.

Note	If the path curve is tangential to the sweep profile, then the sweep will fail.

To sweep an object along a path:

1. Create the polyline you want to use as a path.
2. Select the object you want to sweep, and then select the new polyline.
3. From the menu bar, click **Draw> Sweep> Along Path** or, in the **Draw** ribbon tab, click **Along a path**:

   The **Sweep along path** dialog box appears.



4. Type an angle in the **Angle of twist** text box and choose the desired unit of measure.

   This is the angle the profile will rotate about the sweep path vector as it is swept through the complete path.
5. Type an angle into the **Draft angle** text box and select the desired unit of measure.

   This is the angle to which you want the object's profile, or shape, expanded (positive angle) or contracted (negative angle) as it is swept. Among other purposes, drafts make it easy to remove molded or cast parts from their molds.

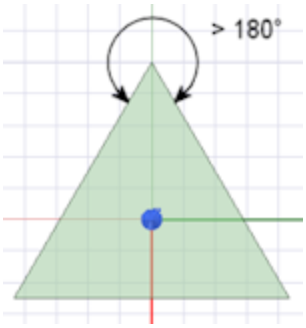
   For 1D objects, the draft angle alters the sweep direction but does not alter the length of the object as it is swept. For example, a 10 mm straight line swept with a 5-degree draft angle does not produce a trapezoidal 2D object. It produces a rectangle with a constant 10 mm width but tilted 5 degrees relative to the specified sweep path.
6. Select one of the following **Draft type** options from the drop-down menu. The draft type instructs the modeler whether to maintain the original object shape and how to transition between adjacent faces. This setting affects the results when sweeping 2D objects into 3D solids or when sweeping lines and curves into 3D sheets.
   - **Extended**
   - **Round**
   - **Natural**

   For a complete description of the behavior of each of these three options, see the Draft Types page.
7. Click **OK**.

   The object is swept along the path. The polyline object used as the path is deleted. The new object has the properties of the original object. The **Properties** dialog box appears, enabling you to modify the object's properties.

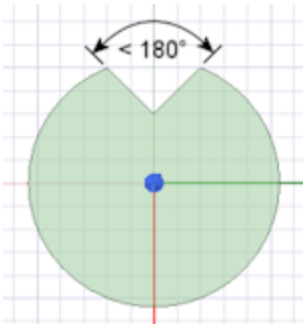
Draft Type Options

In many cases, the **Draft type** setting may have little to no effect on the resultant geometry of a swept object unless a *positive, non-zero draft angle* is specified. Consider a *sharp corner* feature. For the purpose of this discussion let's define a sharp corner as any protruding feature where the *external* angle between the adjacent faces is greater than 180-degrees:



When this type of feature is expanding, the corner can gradually become rounded. However, if the same corner is contracting, the radius is already zero at the start of the sweep and cannot contract or decrease further. Therefore, no rounding occurs in the latter case.

Now, consider a *groove* in the perimeter of a 2D object to be swept. Let's define a groove as any indented feature, where the *external* angle between the adjacent faces is less than 180-degrees:



A groove will grow in size as the object is swept with a *negative* draft angle specified. In this case, progressive rounding of the corners is possible.

You may need to experiment with the three draft type options to determine what the effect will be for nontrivial object shapes. The behavior of each option can differ according to the shape of the 2D object, the draft angle, and whether it is swept along a vector, a path, or about an axis.

The following three **Draft type** options are available:

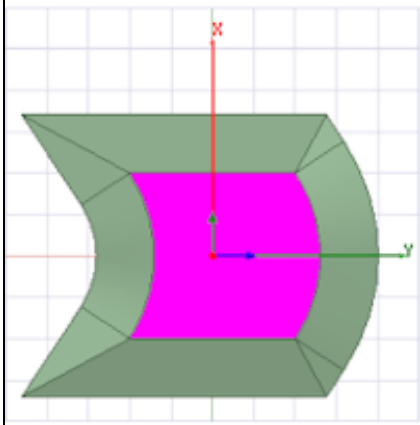
- **Natural:** This draft type generally maintains the shape of the original object as it is swept along the specified direction. That is, all sharp corners remain sharp, producing sharp edges in the sweep direction. There are exceptions to this general behavior. Rounding of certain corners may occur if required to accommodate sides with complex shapes (for example, a complex side might be split into a combination of planar, cylindrical, helical, or conic-section faces).
- **Extended:** This option maintains the shape of the original object as it is swept along the specified direction. Sharp corners of the original object remain sharp corners along the entire swept direction. However, the side faces may be split into three separate tangential faces, as needed and as described in the preceding bullet.

- **Round:** This option rounds all sharp corners of the original object as it is swept along the specified direction.

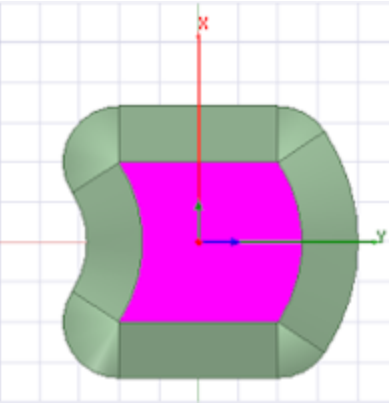
Examples

The images below compare the results of sweeping an object using each of the available **Draft type** options. The first six images are of an object swept with a positive draft angle defined. The final three images are of a different object swept with a negative draft angle defined. For all examples, the original 2D object is highlighted in magenta.

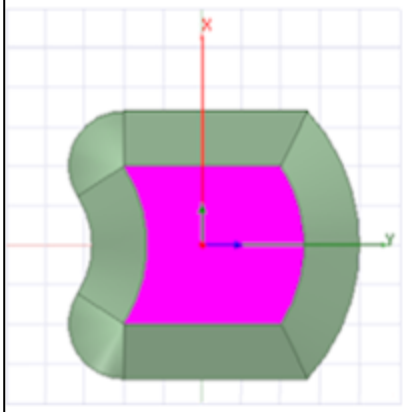
Sweep Along Vector (Positive Draft Angle):



1a: Extended

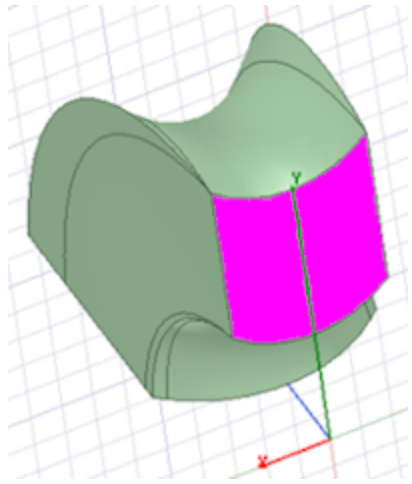


1b: Round

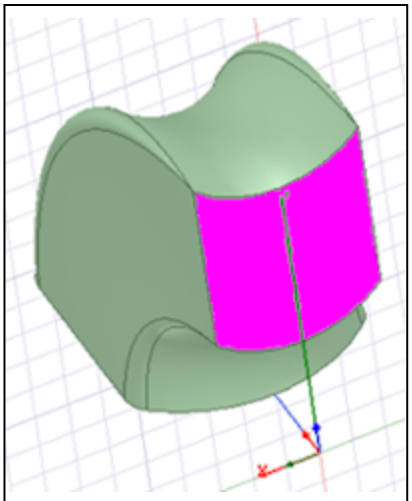


1c: Natural

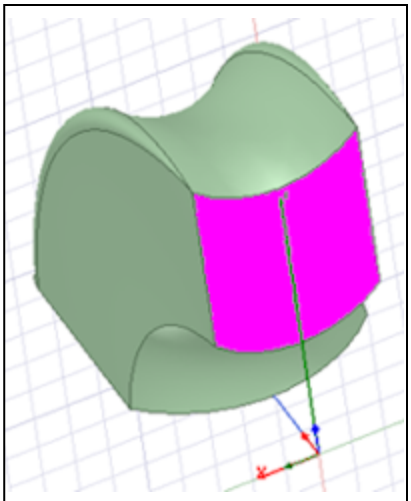
2. **Sweep About Axis** (Positive Draft Angle):



2a: Extended

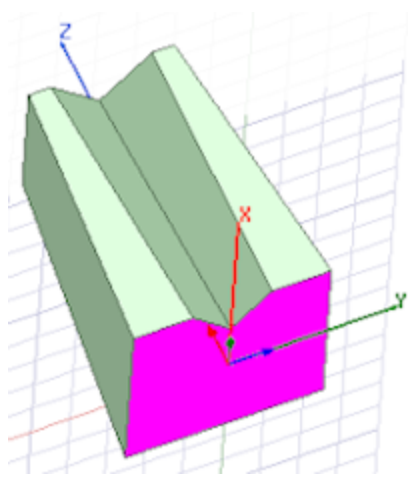


2b: Round

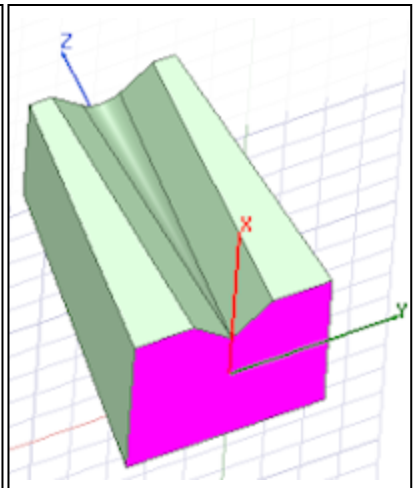


2c: Natural

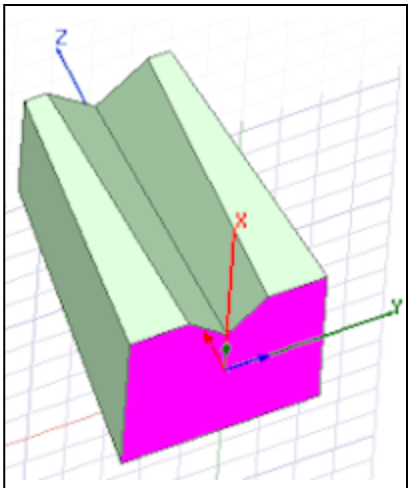
3. **Sweep Along Vector** (Negative Draft Angle):



3a: Extended



3b: Round



3c: Natural

Observations:

- **Extended Option:** Notice how two of the four sides of example **1a** have been divided into three faces each (the sides generated by sweeping curved edges). However, the other two sides (those generated from straight edges) have not been divided.
  For example **2a**, all four sides have been divided into three faces each. Even the straight sides generate a complex surface due to the combination of a positive draft angle and a circular sweep path.
  For example **3a** (negative draft angle), no face is split and no edge is rounded.
- **Round Option:** In both examples **1b** and **-b**, all corners of the original 2D object have been progressively rounded (with the radius increasing from zero at the plane of the original object to a maximum radius at the end of the sweep).
  For example **3b** (negative draft angle), only the corner of the groove has been rounded. The indented feature is the only one increasing is size along the sweep direction.
- **Natural Option:** In both examples **1c** and **2c**, the corners adjacent to the convex curved face remain sharp. However, the corners adjacent to the concave curved face are progressively rounded in both cases (with the radius increasing from zero at the plane of the original object to a maximum radius at the end of the sweep).
  For example **3c** (negative draft angle), no face is split and no edge is rounded.

Sweeping Faces Along Normal

To create a new object by sweeping select 3D object's face a specified distance in a direction normal to its original plane, use the **Modeler>Surface>Sweep Faces Along Normal** command. Note that the adjoining faces will not be sheared or bent.

This command is useful for extruding faces, resizing holes, and removing rounded corners.

To sweep selected object faces in a normal direction:

1. Click **Select Faces** on the shortcut menu.
2. Select the faces of the object you want to sweep.
3. Click **Modeler>Surface>Sweep FacesAlong Normal**.
4. The **Sweep FacesAlong Normal** dialog box appears.
5. Type the distance you want to sweep the object face from its origin.
6. Click **OK**.

   The face is swept the distance you specified to create a new object.

Related Topics

Moving Faces Along the Normal

Thicken Sheet

To thicken a sheet object to make a 3D object:

1. Select the sheet.
2. Click **Modeler>Surface>Thicken Sheet**.

   The **Thicken Sheet** dialog appears.
3. Specify the thickness by typing in the field.
4. Specify the units by selecting from the drop down menu.
5. If you want to thicken both sides, use the check box.
6. Click **OK**.

   The dialog closes and the sheet is changed into a 3D object of the desired thickness.

Wrap Sheet Command

You can use **Modeler>Surface>Wrap Sheet** command to wrap a sheet object around a suitable 3D object. The sheet object does not have to be in contact with the 3D object. It should have smaller dimensions than the 3D object.

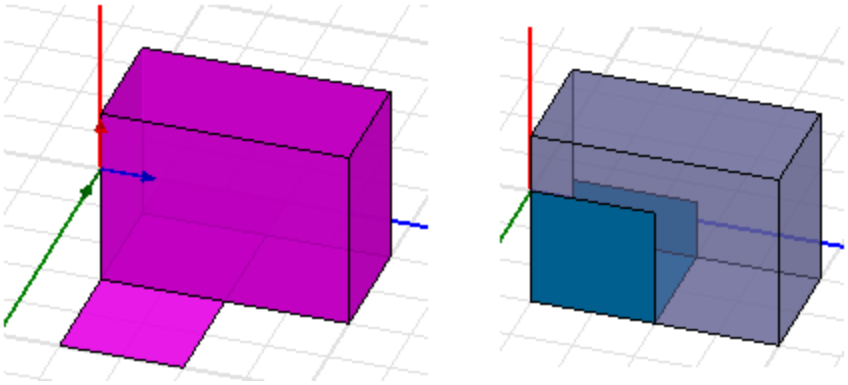
When the sheet to be wrapped and the target body do not touch, the wrap sheet command chooses among several different methods for bringing them into contact.

- If an edge from the sheet body and the target body are close enough and at a small angle (10 degrees), the edges are brought into alignment before the wrap.
- If a vertex from the sheet body and an edge on the target body are close enough, the vertex is moved to the target edge.
- If the previous criteria are not matched, a vertex from the sheet body is moved to the closest face of the target body.

If both the sheet to be wrapped and the target face are both planar and have the same or opposite normal, the closest vertex from the sheet body is moved to the target face. No movement to or alignment of edges is performed.

If the sheet body has multiple lumps, a single rectangle is fitted around all of the lumps. The rectangle is moved to the target surface using the above procedure.

If the sheet object does not overlap the corners of the 3D object, the wrap is straightforward, as shown in the figure.

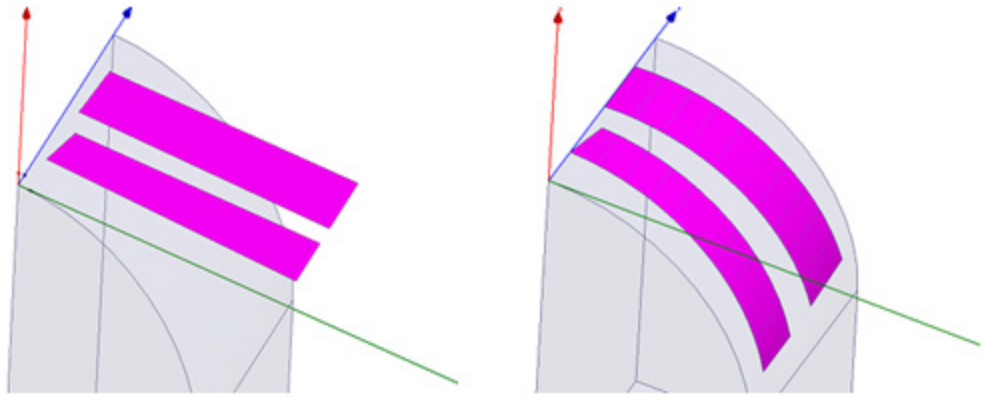


If you delete the 3D object, the wrapped sheet retains the form it took when wrapped.

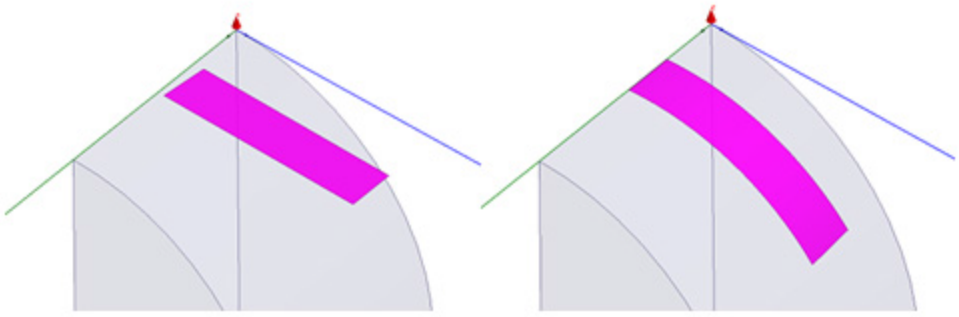
A sheet object that overlaps corners may not wrap in straightforward fashion, depending on both the angle(s) involved, and the sheet object. While it is possible, it is not recommended.

Examples:

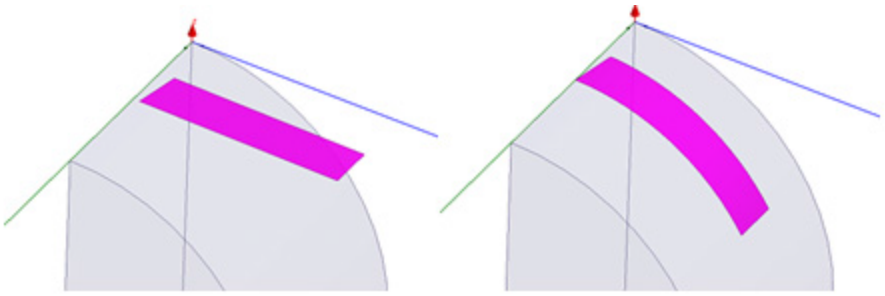
The following figure illustrates wrapping on a target surface with the snap to an edge. The surface will snap to an edge if close relative to length of edge. If the angle between the edge is less than 10 degrees, the edge to be wrapped will be aligned to the target body. In the example on the left, the lower sheet has an edge angle more than 10 degrees. The vertex will be snapped to the edge. The top sheet in the picture has an edge angle of less than 10 degrees. The vertex is snapped to the target edge and the sheet edge is aligned to the target. To avoid snapping to an edge or edge alignment the wrap sheet can be put into contact with the target surface. In that case the sheet will vertex will not be moved to target edge unless it is within modeling tolerances.



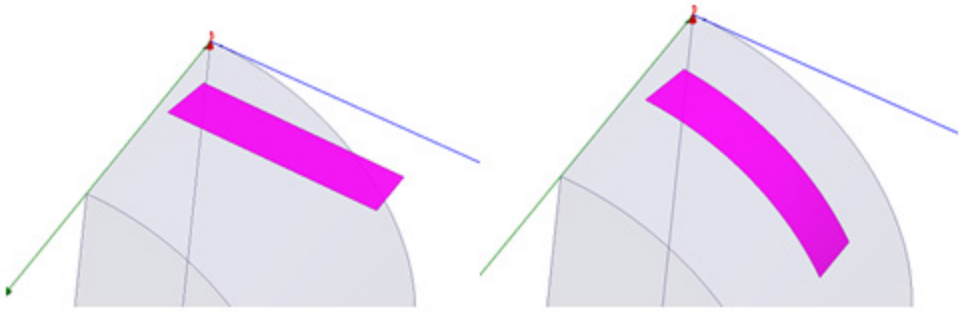
The following figure shows the situation when you move the edge to the edge before wrapping.



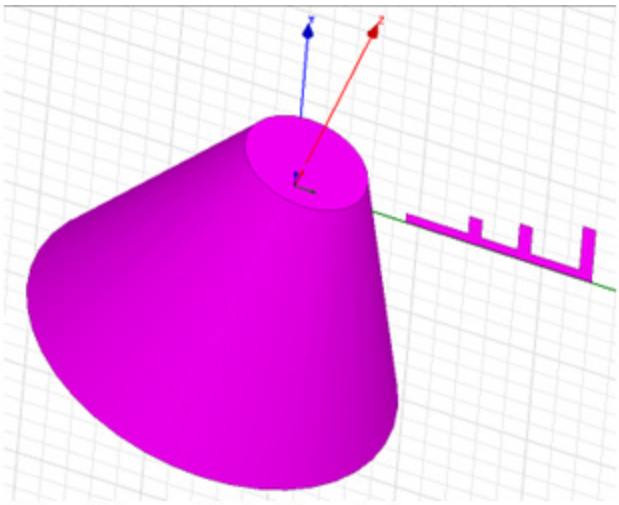
The following figure shows the situation when you move a vertex to an edge before wrapping.



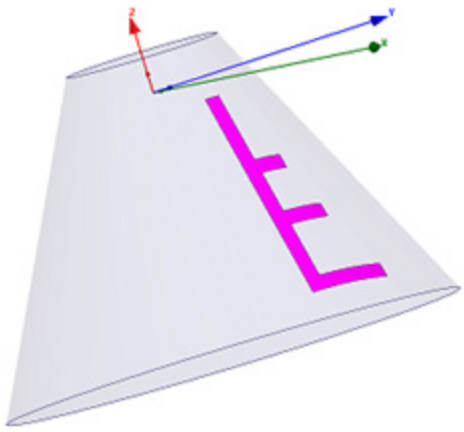
The following figure shows the situation where you move a vertex to a face before wrapping.



The figure below illustrates wrapping sheet onto cone. The sheet does not have line contact.

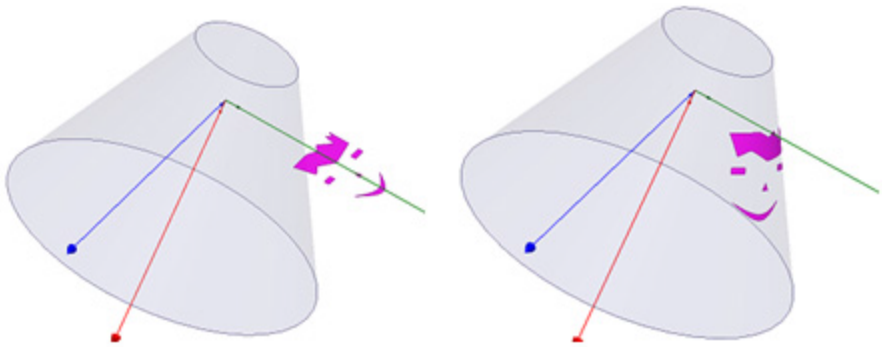


Picture 1: Use case 1 before Wrap sheet

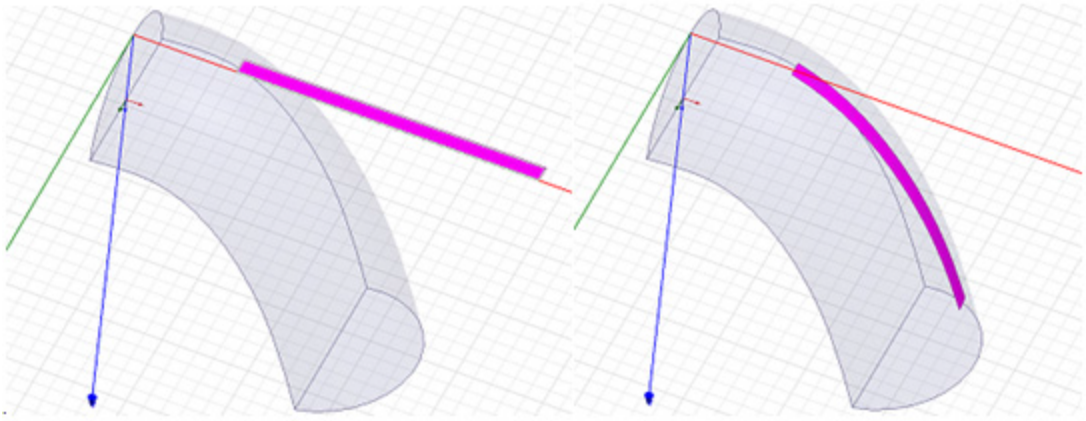


Picture 2: Use case 1 after Wrap sheet.

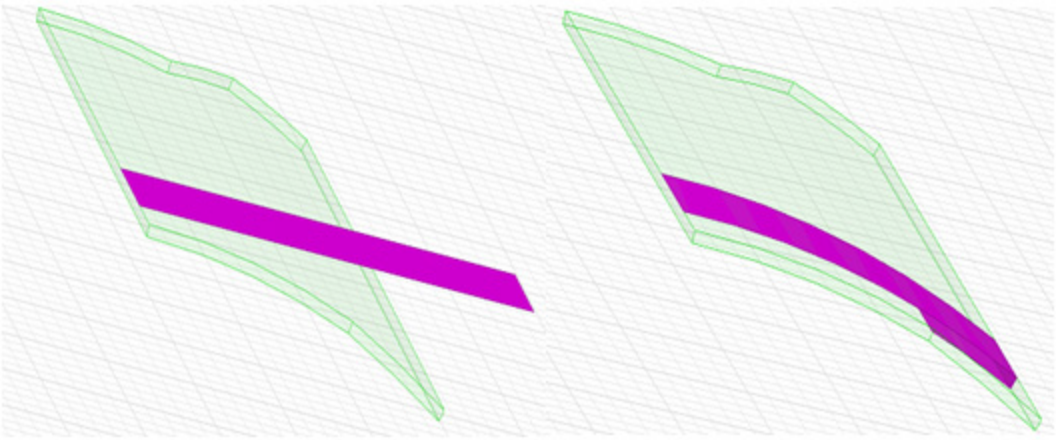
The following figure shows the situation when you move a multilump sheet to touch the target body.



The next figure illustrates wrapping to a non-developable surface. The area of sheet changed from 0.7000mm^2 before wrap to 0.6986 mm^2 after.



The following figure illustrates wrapping on a target surface which has tolerant edges. The tolerant edge is at the beginning of the wrap. The tolerant edge is not evident visually.



Limitations/Error Conditions:

- Wrapping on non-developable surfaces creates distortion in lengths.
- Wrapping is not allowed across curved boundaries between faces.
- Working with tolerant edges will work only if the gaps can be tightened sufficiently for the operation.
- Wrapping on the pole of a surface (such as sphere or cone) is not supported. A warning is given: WrapSheet: Wrapping sheet contacted pole of target surface. Recommend moving sheet and splitting out pole from target surface.
- The choice of method for how the sheet body is moved to touch the target face is not directly selectable by the user. If the sheet to be wrapped and target body touch, then the sheet body is only rotated at point of contact.

To wrap a sheet object:

1. Create a sheet close to an appropriate 3D object.
2. Select both objects.
3. Click **Modeler>Surface>Wrap Sheet.**

   The sheet object wraps around the 3D object. You can select the wrapped sheet object and the 3D object separately, and assign properties separately.

   If the object cannot wrap, the Message window contains a warning and description.

   You can wrap multiple sheets on the same 3D object.

   If you delete the 3D object, the wrapped sheet retains the form it took when wrapped.

You can use the **Tools>Options>Modeler** Options to automatically perform a **Modeler>Boolean>Imprint** command after performing a **Wrap.**

Related Topics

Imprinting an Object

Modeler Options: Operation Tab

Covering Lines

To cover a closed 1D polyline object with a face, use the **Modeler>Surface>Cover Lines** command. The polyline object becomes a 2D sheet object.

To convert a polyline object to a sheet object:

1. Select the closed polyline object you want to cover.
2. Click **Modeler>Surface>Cover Lines**.

   The object is now covered. It is now a 2D sheet object that can be swept to form a 3D solid object.

Note	If you want the modeler to automatically cover all closed polyline objects you draw, including circles, ellipses, rectangles, and regular polygons, select the **Automatically cover closed polylines** option in the **Modeler Options** dialog box. A closed polyline object can also be created by using boolean unite operations on two or more polylines.

Covering Faces

To cover object faces, the faces must be united into a 3D sheet object. To cover the face of a 2D or 3D object, use the **Modeler>Surface>Cover Faces** command.

Covering the face of an open 2D sheet object that had previously been uncovered results in a 3D solid object. For example, for a box, when you select and uncover a face, the solid box becomes a sheet with five faces. When you then select that sheet body box and use the **Cover Faces** command, the box becomes a solid again with six faces.

To cover the faces of objects:

1. Select the faces of the objects you want to cover.
2. Click **Modeler>Surface>Cover Faces**.

   The object faces are now covered.

Uncovering Faces

Uncover a surface of a 3D object using the **Modeler>Surface>Uncover Faces** command. Uncovering the surface of a 3D solid object results in an open 2D sheet object.

To uncover the face of a 3D object:

1. Switch to face selection mode: Click **Edit>Selection Mode>Faces**.
2. Select a face of the object you want to uncover.
3. Click **Modeler>Surface>Uncover Faces**.

   The selected face is uncovered, leaving an open face on the object.

Note	You can uncover one face of a 3D object at a time. If you select multiple faces, only the first face will be uncovered.

Detaching Faces

The **Modeler>Surface>Detach Faces** command enables you to remove the face of a 3D object, resulting in two separate objects.

To detach the face of an object:

1. Switch to face selection mode: Click **Edit>Selection Mode>Faces**.
2. Select the face of the object you want to detach. You can select multiple faces to detach.
3. Click **Modeler>Surface>Detach Faces**.

   The selected face is now detached, resulting in two 2D sheet objects.

Detaching Edges

The **Modeler>Edge>Detach Edges** command enables you to remove an edge of a wire object, resulting in two separate wire objects.

To detach an edge of an object:

1. Switch to edge selection mode: Click **Edit>Selection Mode>Edges**.
2. Select the edge of the object you want to detach. You can select multiple edges to detach.
3. Click **Modeler>Edge>Detach Edges**.

   The selected edge is now detached, resulting in multiple wire objects.

Note	Only edges from wire bodies can be used in a detach edge operation.

Creating a Cross-Section

You can take a cross-section of a 3D object to create a new 2D object. This is done using the **Modeler>Surface>Section** command.

Use this command to create cross-sections of 3D objects on the xy, yz, or xz plane. The cross-sections are created as 2D closed polyline objects.

To create a cross-section of an object:

1. Make sure the working coordinate system you want to use for the cross-sectioning plane is set.
2. Select the object from which you want to create a cross-section.
3. Click **Modeler>Surface>Section**.
4. Select the section plane you plan to use to divide the object: **XY**, **YZ**, or **ZX**.
5. Click **OK**.

   A closed polyline object is created from the object that was sliced by the selected plane. The original, sectioned object is unmodified.

Related Topics

Setting the Working Coordinate System

Connecting Objects

Use the **Modeler>Surface>Connect** command to perform the following operations:

- Connect two or more 1D polyline objects. Maxwell will modify the first polyline you select to be a 2D sheet object that connects to the second and any subsequently selected polylines. The second and subsequent polylines selected are deleted.
- Connect two or more 2D sheet objects. Maxwell will modify the first 2D object you select to be a 3D solid object that connects to the second and any subsequently selected objects. The second and subsequent objects selected are deleted.

To connect objects:

1. Select the objects you want to connect.
2. Click **Modeler>Surface>Connect**.

   A new object is created that connects the objects you selected. The first object you selected was modified to create the new object and all subsequently selected objects were deleted.

Moving Faces

You can move the faces of a 3D object in a normal direction using the **Modeler>Surface>Move Faces** commands. Moving object faces enables you to resize, reshape, or relocate an object.

Related Topics

Moving Faces Along the Normal
Moving Faces Along a Vector
Offsetting Objects
Moving Edges Along the Normal

Moving Faces Along the Normal

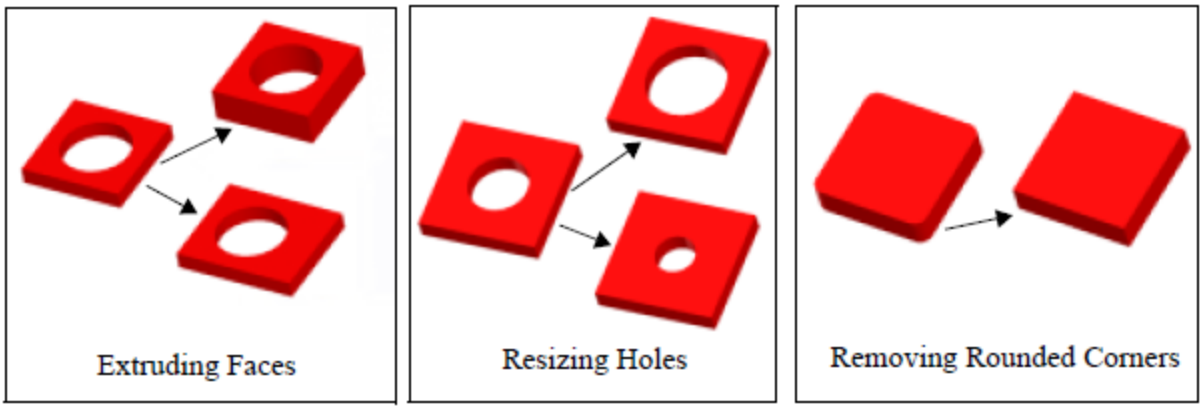
To move a 3D object's face a specified distance in a direction normal to its original plane, use the **Modeler>Surface>Move Faces>Along Normal** command. The faces that adjoin the original face are extended or shortened along their own planes to meet the new face. Note that the adjoining faces will not be sheared or bent.

This command is useful for extruding faces, resizing holes, and removing rounded corners, as shown below.

To move an object face in a normal direction:

1. Click **Select Faces** on the shortcut menu.
2. Select the face of the object you want to move.
3. Click **Modeler>Surface>Move Faces**>**Along Normal**.
4. The **Move facesalong normal** dialog box appears.
5. Type the distance you want to move the object face from its origin.
6. Click **OK**.

   The face is moved the distance you specified.



To move every face of an object normal to its surface, use the **Edit>Arrange>Offset** command.

Related Topics

Moving Faces Along a Vector

Offsetting Objects

Moving Edges Along the Normal

Moving Faces Along a Vector

To move the faces of a 3D object a specified distance along a vector use the **Modeler>Surface>Move Faces>Along Vector** command. Each selected face is moved along the vector, normal to its original plane. The faces that adjoin the original face are extended or shortened along their own planes to meet the new face. Note that the adjoining faces will not be sheared or bent.

This command is useful for relocating holes in an object, as shown below.

To move an object face along a vector:

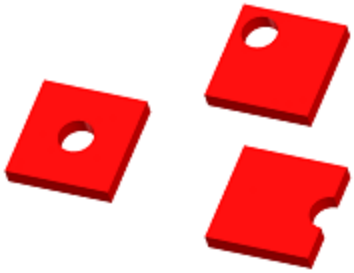
1. Click **Select Faces** on the shortcut menu.
2. Select the face of the object you want to move.
3. Click **Modeler>Surface>Move Faces**>**Along Vector.**
4. Specify the vector along which the face is to be moved:
   a. Select an arbitrary anchor point in one of the following ways:
      - Click the point.
      - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.

   Any point in the drawing region can be selected; however, selecting an anchor point on the object's edge or within the object makes it easier to select the vector.

   b. Select a second point in one of the following ways:
      - Click the point.
      - Type the coordinates of a point relative to the anchor point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   This point defines the direction and distance from the anchor point to move the face.

   The face is moved along the vector you specified.



To move every face of an object normal to its surface, use the **Edit>Arrange>Offset** command.

Related Topics

Moving Faces Along the Normal
Offsetting Objects
Moving Edges Along the Normal

Moving Edges Along Normal

To move a 2D object's edge a specified distance in a direction normal to its original plane, use the **Modeler>Edge>Move Edge** command. The edge is extended or shortened along its own plane. Note that the adjoining faces will not be sheared or bent. The edge can be on a rectangle, an ellipse, a circle, a regular polygon, or an equation based surface.

This command is useful for extending or shrinking faces and resizing holes.

To move an object edge in a normal direction:

1. Click **Select Edge** on the shortcut menu.
2. Select the edge of the object you want to move.
3. Click **Modeler>Edge>Move Edge**.
   The **Move FacesAlong Normal** dialog box appears.
4. Type the distance you want to move the object face from its origin.
5. Click **OK**.
   The edge of the object is moved based on the value you specified.

Related Topics

Select Edges.
Moving Faces Along the Normal
Moving Faces Along a Vector
Offsetting Objects

Uniting Objects

To join two or more objects into one object, use the **Modeler>Boolean>Unite** command. The new object has the name, color, boundary, and material assignment of the first object selected. The objects are united at the point of intersection.

You can unite objects that do not touch.

To unite two or more objects:

1. Select the objects you want to join.
2. Click **Modeler>Boolean>Unite**  ..
   The objects are united.

Note	By default, the objects being joined to the first object selected are *not* preserved for later use. If you want to keep a copy of the objects being joined to the first object selected, do one of the following: • Copy the objects, and then paste them back into the design after uniting them. • Select **Clone before unite** in the **Modeler Options** dialog box. This option instructs the modeler to always keep a copy of the original objects being joined.

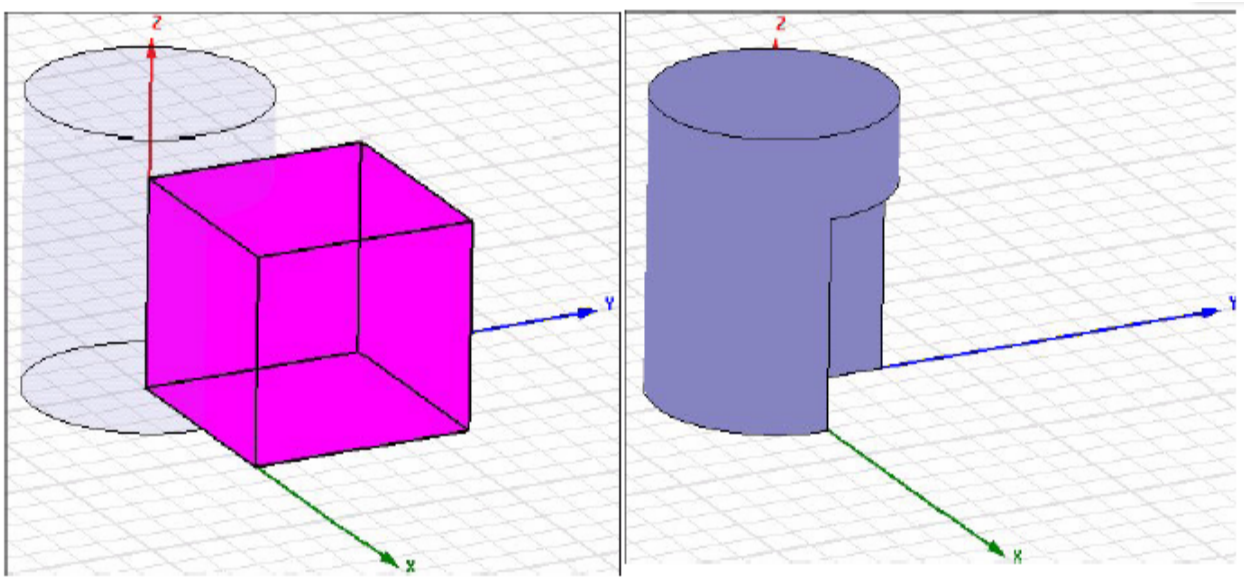
Subtracting Objects

1. Select the object from which you want to subtract other objects.
2. Hold down the **Ctrl** key and select the objects you want to subtract.
3. Click **Modeler**>**Boolean**>**Subtract**  .

   The **Subtract** dialog box appears.

   Objects listed in the **Tool Parts** list are subtracted from the object or objects listed in the **Blank Parts** list.
4. Optionally, select an object name in either list, and use the left and right arrow buttons to move the object name to the opposite list.
   - Alternatively, type the name of object you want to subtract in the empty box below the **Tool Parts** list, and then type the name of the object from which you want to subtract it in the empty box below the **Blank Parts** list.
5. Optionally, select **Clone tool objects before subtract**. This instructs Maxwell to always keep a copy of the original objects being subtracted.
6. Click **OK**.

   The new object retains the name, color, and material of the first object selected.



An intersecting box and cylinder.

A box subtracted from a cylinder.
The cylinder was selected first.

Note	By default, the objects being subtracted from the first object selected are *not* preserved for later use. If you want to keep a copy of the objects being subtracted from the first object selected, do one of the following: • Copy the objects, and then paste them back into the design after subtracting them. • Select **Clone before subtract** in the **Modeler Options** dialog box. This option instructs Maxwell to always keep a copy of the original objects being subtracted.

Creating New Objects By Intersecting Objects

To create a new object from the intersection of two or more objects, use the **Modeler>Boolean>Intersect** command.

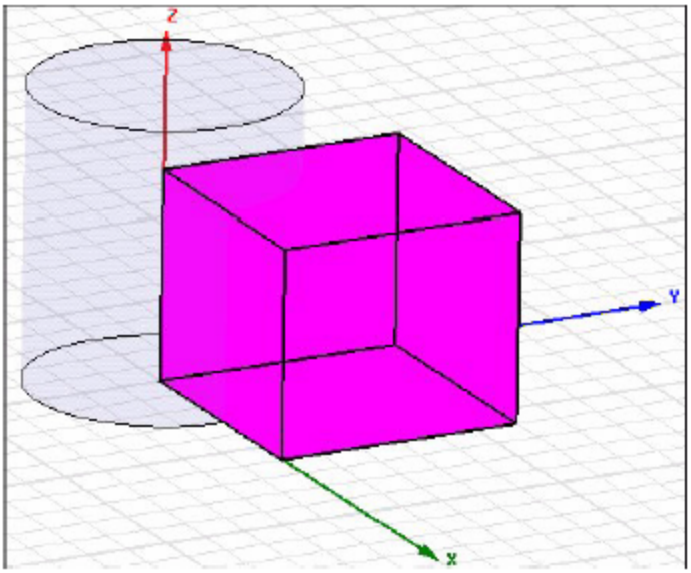
To create an object from an intersection:

1. Select the objects from which you want to take the intersection.

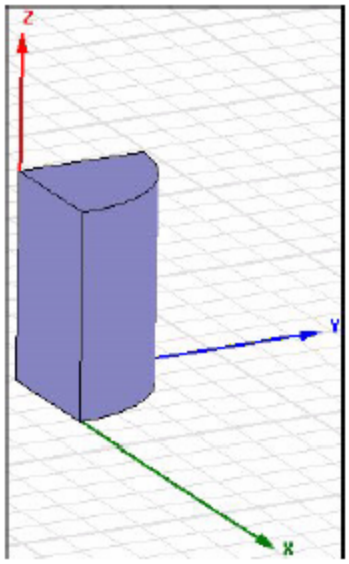
Warning	If the objects you selected do not overlap, the result is a null object, and all objects vanish.

2. Click **Modeler**>**Boolean**>**Intersect**  .

   The original objects vanish, leaving only the new object that was formed from their intersection.



An intersecting box and cylinder.



Object formed from the intersection of the box and cylinder.

Note	By default, the original intersecting objects are *not* preserved for later use. If you want to keep a copy of the objects that intersect the first object selected, do one of the following: • Copy the objects, and then paste them back into the design after creating the new object from the intersection. • Select **Clone before intersect** in the **Modeler Options** dialog box. This option instructs the modeler to always keep a copy of the original objects that intersect the first object selected.

Creating an Object from a Face

The **Modeler>Surface>Create Object From Face** command copies a selected face, resulting in a new 2D sheet object.

To create a new object from a face:

1. Right-click in the modeler window, and select **Select Faces** on the shortcut menu.
2. Select the object face you want to copy. If you select multiple faces, each becomes a new object.
3. Click **Modeler**>**Surface>Create Object From Face**.

   The face is copied, resulting in a new 2D sheet object.

Hint	This command is useful for assigning a boundary to the intersection of two faces. To do this, first select the faces, and then create an object from them using the procedure

	above. Next, make sure the **Clone before intersect** option is clear in the **Modeler Options** window, and then use the **Modeler>Boolean>Intersect** command to modify the object so that it includes only the intersection of the two faces. Then assign the boundary to the new object.

Related Topics

Assigning a Cross Section and Dimension to a Polyline

Creating an Object from an Edge

Creating an Object from an Edge

The **Modeler>Edge>Create Object From Edge** command copies a selected edge, resulting in a new 2D sheet object.

To create a new object from an edge:

1. Right-click in the modeler window, and select **Select Edges** on the shortcut menu.
2. Select the object edge you want to copy. If you select multiple edges, each becomes a new object.
3. Click **Modeler>Edge**>**Create Object From Edge**.

The edge is copied. The resulting object appears in the history tree as a line object.

Related Topics

Assigning a Cross Section and Dimension to a Polyline

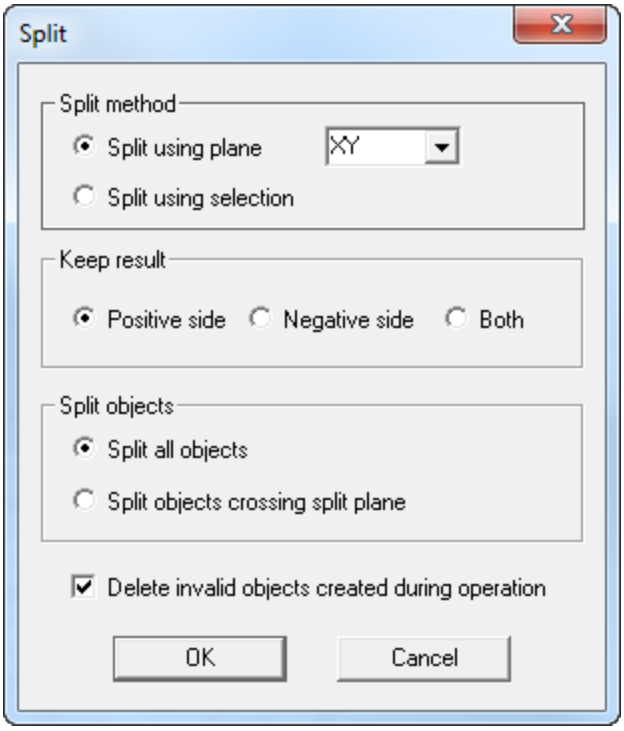
Creating an Object from a Face

Splitting Objects

To delete parts of an object that lie on the xy, yz, or xz plane, use the **Modeler>Boolean>Split** command.

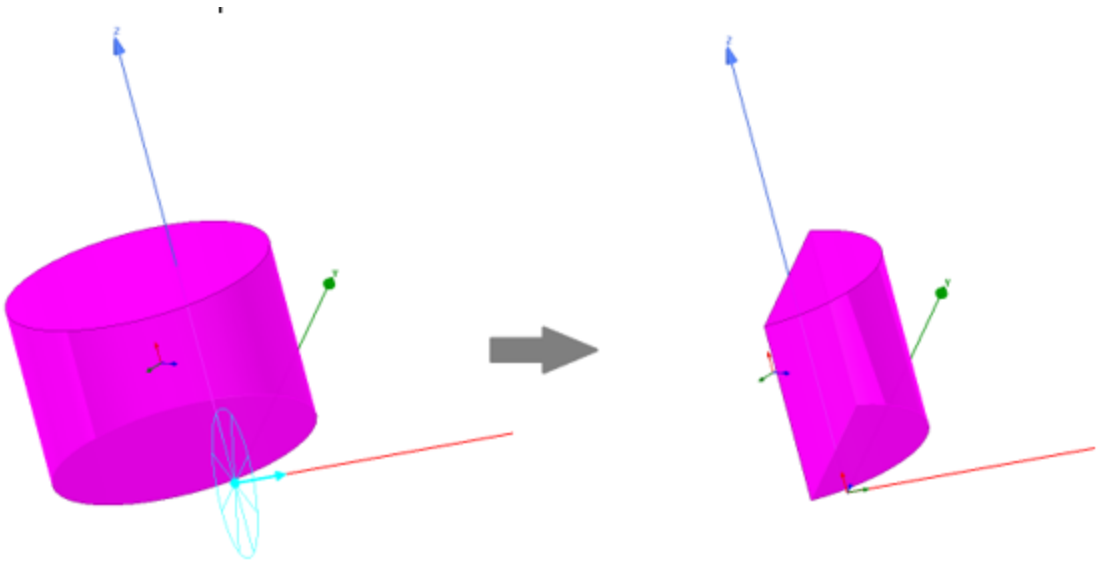
1. Select the object you want to split. You can select more than one.
2. Click **Modeler>Boolean>Split**  .

   The **Split** dialog box appears.

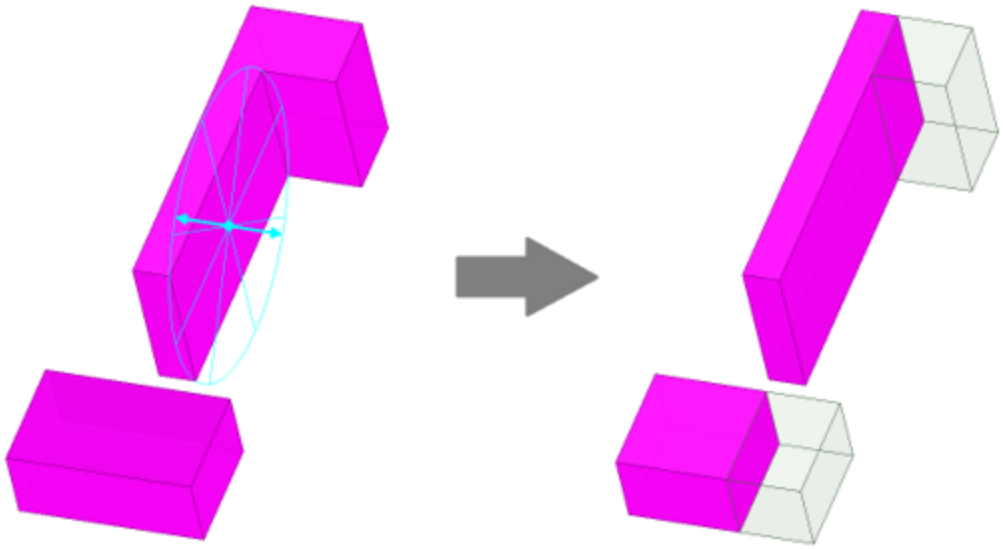


3. Select **Split using plane**, and then select **XY**, **YZ**, or **XZ** as the **Split plane** or click **Split using selection** and select a planar face or arc edge.
4. Select one of the following **Keep result** options to specify which object fragments you want to keep:
    - **Positive side**- keep objects on the positive side of the selected plane.
    - **Negative side** - keep objects on the negative side of the selected plane.
    - **Both** - all pieces on both sides of the plane.

    The split plane is shown with a blue wheel with spokes as seen in the picture below left. A blue arrow, in case to the right, indicates which side of the split plane is to be kept. Once the proper selections have been made press OK. In this example the positive side of YZ (green and blue axes) plane results was kept. The picture on the left is before the split. The picture on the right is the result for the split command.



If you select **Split using selection** you can choose a planar face or edge arc to define the split plane. You can only select one entity. Face or edge to define plane need not belong to one of objects selected for split operation - it could be any face or edge from the model. You can position, scale, or view the model as needed to make the selection easier. The following example shows the operation with keep result for both sides. (Notice the arrow is in both directions for the split plane graphics.) The picture on the left is before the split. The picture on the right is the result for the split command.



5. Select one of the following **Split objects** options:
   - **Split entire selection** –Select this option if you do not want to preserve objects that are not crossing the split plane and still part of the selection.

- **Split all objects crossing split plane** – This option allows you to identify selected objects that do not cross the split plane and ignore them for the operation. For a multiple selection, only those objects that cross the split plane are split; others are kept intact. By design, splits in existing designs from previous versions are not changed.

6. If you choose split entire selection, it could create invalid objects if the split plane does not cut any of the selected objects. You can use the option **Delete invalid objects created during operation** to delete these invalid objects.
7. Click **OK**.

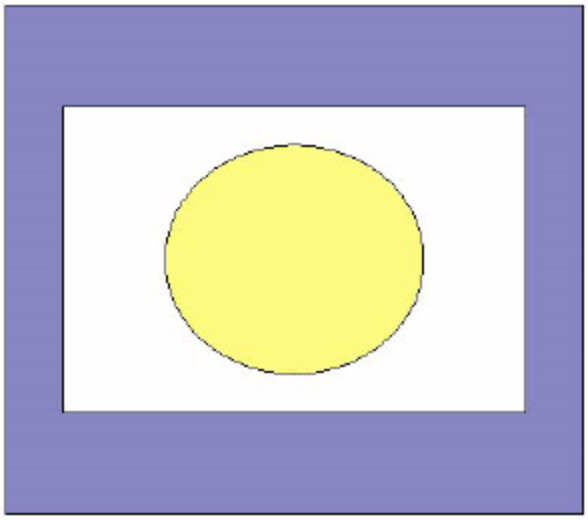
   The objects are divided as specified.

Separating Bodies

To separate an object with multiple lumps into individual bodies:

1. Select the object you want to separate.
2. Click **Modeler>Boolean>Separate Bodies**.

   The object is separated.



This figure shows two separate bodies, each with one lump, that were created from one object.

Converting Polyline Segments

A polyline is a single object that includes any combination of straight line, arc line, or spline segments. You can convert a polyline segment from one type to another. The following conversions are supported:

- Straight line segments to arc line or spline segments.
- Arc line segments to straight line or spline segments.

- Spline segments to straight line segments.

To convert polyline segments:

1. In the history tree, locate the polyline that contains the segment you want to convert. Expand this part of the history tree.
2. In the history tree, right-click the polyline segment operation you want to change, and then click **Properties**.
   The Properties dialog appears.
3. In the **Properties** dialog box, click in the **Value** box of the **Segment Type** row.
4. Select the desired polyline segment type from the pull-down list.
   The polyline segment you selected is changed to the new type.

Note	Converting an arc line or spline segment to a straight line segment results in two straight line segments; one segment is created between the start point and midpoint, and one segment is created between the midpoint and endpoint.

5. By default, curved surfaces are treated as smooth (True) surfaces. If segmented surfaces are desired, enter an integer number in the **Number of Segments** parameter.
6. Click **OK** to dismiss the properties panel and implement the changes.
   If the changes are not what was expected, undo the change using the **Edit>Undo** command or press **CTRL-Z**.

Related Topics

Assigning a Cross Section and Dimension to a Polyline

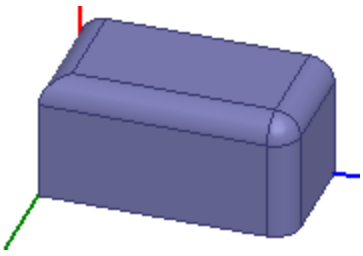
Creating Segmented Geometry

Surface Approximation

Rounding the Edge of an Object (Fillet Command)

Note	The Fillet command may only be applied to edges of 3D solid objects. Therefore, the command is only available when working on a 3D Project.

The fillet command rounds the object at the original edges and vertices. This means that the edges and vertices are going to be replaced by new rounded surfaces, so that the original faces of the object reconnect in a smooth manner.



Vertices are only going to be replaced by new rounded surfaces if all the edges connecting to the original vertex are selected; otherwise, the vertex is preserved but moved (if necessary). The edges are replaced by quarter-cylindrical surfaces, of which the radius can be customized (see the **Fillet Radius** property). Vertices are replaced by more complicated new faces. You can control the setback distance.

The fillet command is disabled if an edge is not selected.

To switch to edge selection mode:

- Right-click the desktop, and select **Selection Mode>Edges** from the shortcut menu.

To round an object's edge:

1. Select the edge you want to change.

   The edge is highlighted, and the **Fillet** command is enabled.
2. Click **Modeler>Fillet**, or click the **Fillet** icon on the **Draw** ribbon.

   The **Fillet Properties** dialog box appears.
3. Enter a value for the **Fillet Radius** in the text field and select units from the drop down menu. The default is millimeters.
4. Enter a value for the **Setback distance**.

   The setback distance controls the shape of the vertex. It is the distance of the cross curve from the vertex at the end of the edge. If it is less than the fillet radius it has no effect. You will get an error if it is greater than the length of the edge.

Note	The setback feature works only on corners where three or more edges meet and only if all edges meeting at the vertex are selected.

5. Click **OK** to apply the change to the edge.

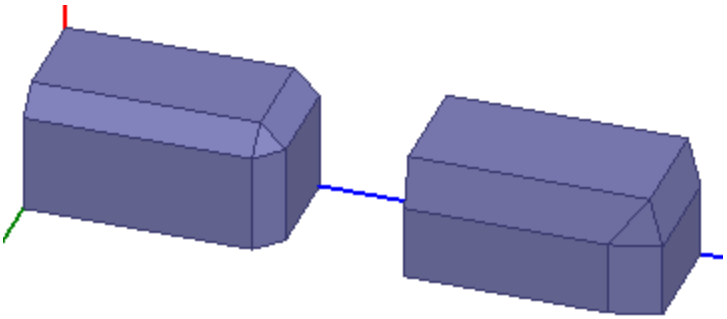
   The dialog closes and the object is rounded by the radius value relative to the edge you selected.

Flattening the Edge of an Object (Chamfer Command)

Note	The Chamfer command may only be applied to edges of 3D solid objects. Therefore, the command is only available when working on a 3D Project.

The chamfer command flattens the edges and vertices of the object. This means that the edges and vertices are replaced by new flat surfaces, so that the original faces of the object reconnect through the newly introduced flat surfaces. Vertices are replaced by new flat surfaces only if all the edges connecting to the original vertex are selected; otherwise, the vertex is preserved but moved (if necessary).

Chamfers can be symmetric, or asymmetric.



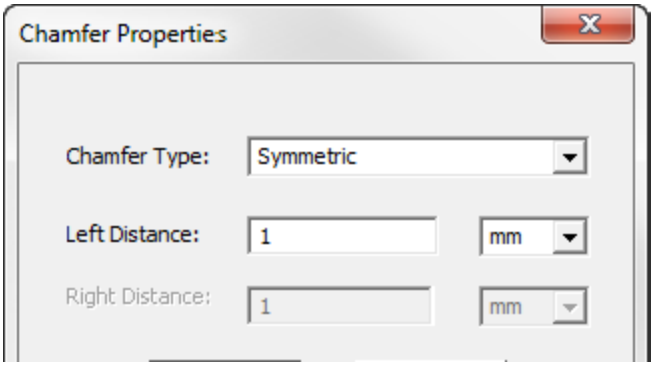
The chamfer command is disabled if an edge is not selected.

To switch to edge selection mode:

- Right-click the desktop, and select **Selection Mode>Edges** from the shortcut menu.

To flatten an object's edge.

1. Select the edge you want to change.
   The edge is highlighted, and the **Chamfer** command is enabled.
2. Click **Modeler>Chamfer** or click the **Chamfer**icon on the **Draw** ribbon.
   The **Chamfer Properties** dialog box appears.

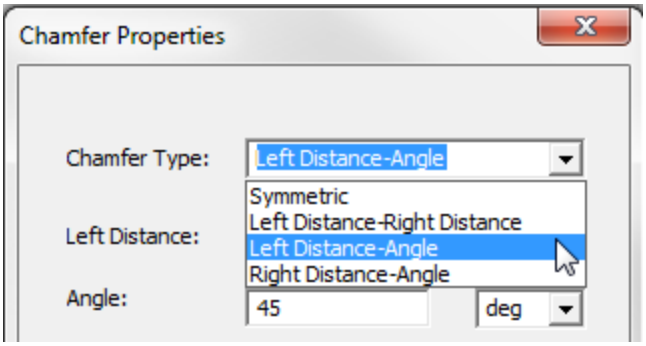


3. Type a value in the **Left Distance** text box, and select the units from the pull-down list.
4. Click **OK** to apply the change to the edge.
   The **Chamfer Properties** dialog box closes, and the object is flattened by the radius value relative to the edge you selected.

To flatten an object's edge for an asymmetric chamfer:

1. Select one or more edges and click **Modeler>Chamfer** to open the **Chamfer Properties** dialog.

2. Use the Chamfer type drop down menu to select the type:



You control an asymmetric chamfer by selecting a type that defines the chamfer asymmetry as Left-Distance Right Distance, as Left Distance-Angle, or as Right Distance Angle. Notice that red directional arrows on the selected edges provide the direction against which left distance and right distance are determined.

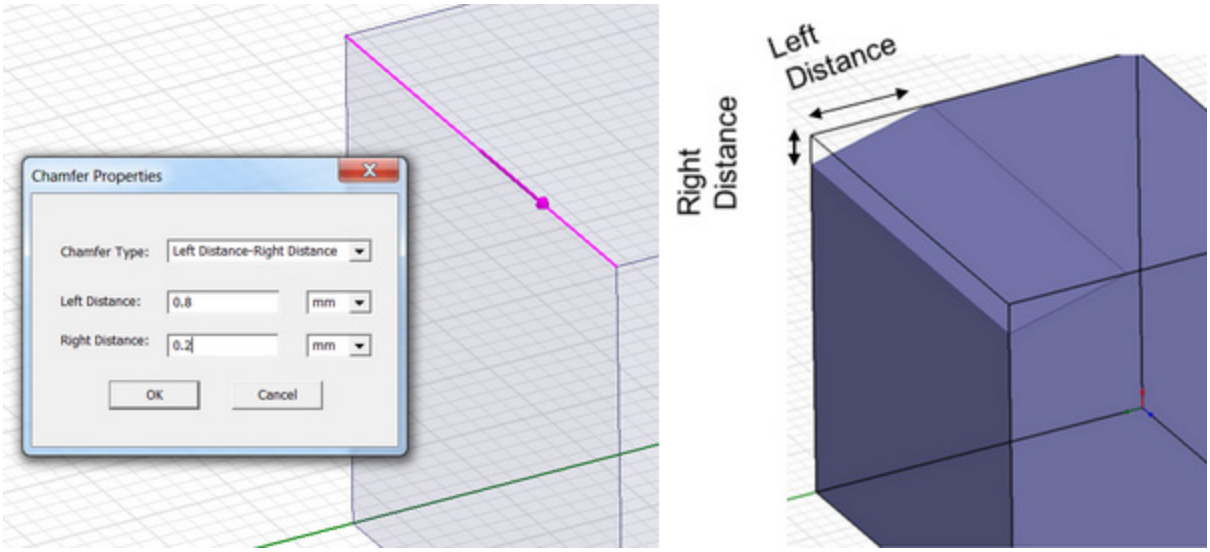


Figure 8-2 Left Distance-RightDistance

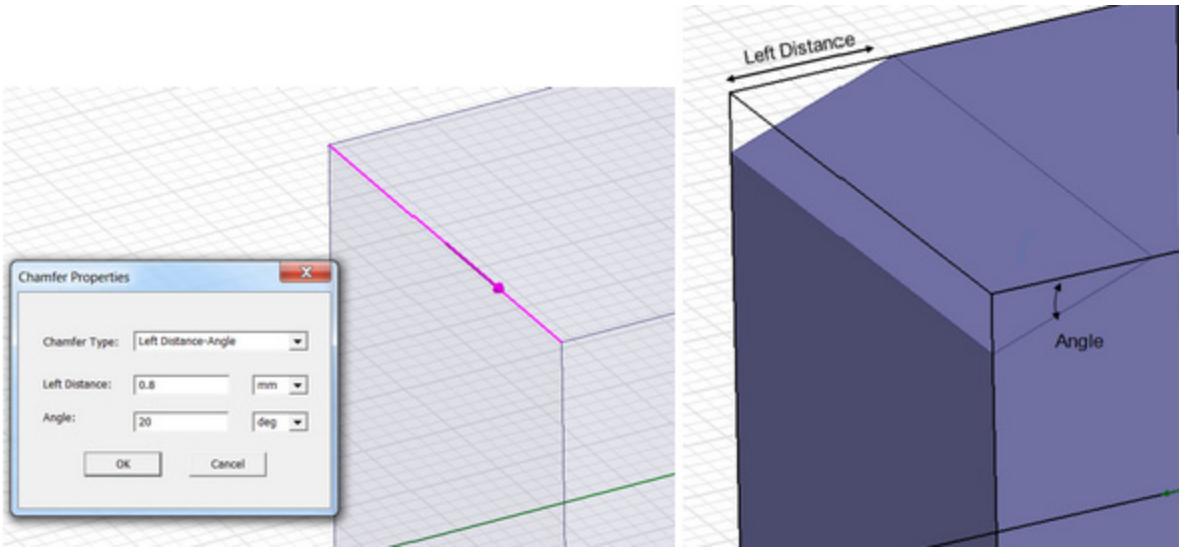


Figure 8-3 Left Distance Angle

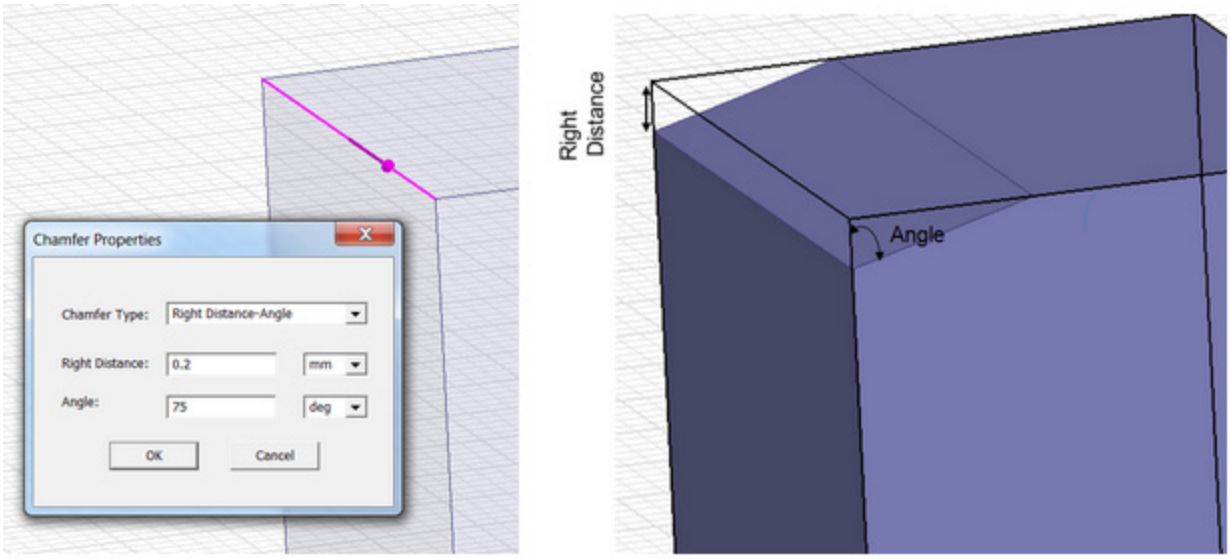
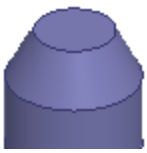


Figure 8-4 Right Distance-Angle

Only symmetric and left distance-right distance chamfers are supported for edges formed by a curved surface.

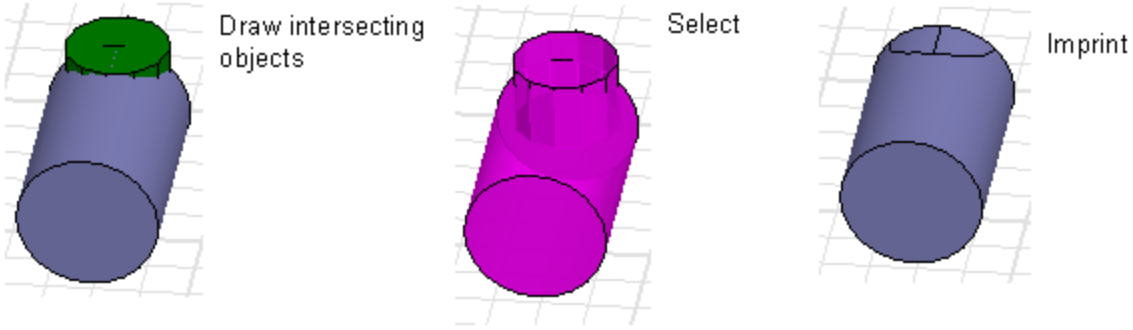


Angle-distance chamfers are not supported in such cases and do not appear on the Chamfer Type drop down menu.

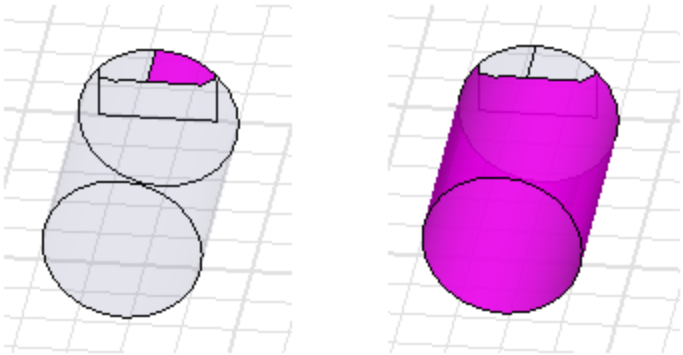
3. Click **OK** to apply the chamfer to the selected edges.

Imprinting an Object

The **Boolean>Imprint** command lets you imprint the geometry of one object upon another. For example, you could draw a polyhedron intersecting a cylinder, and then imprint the intersecting lines on the cylinder.



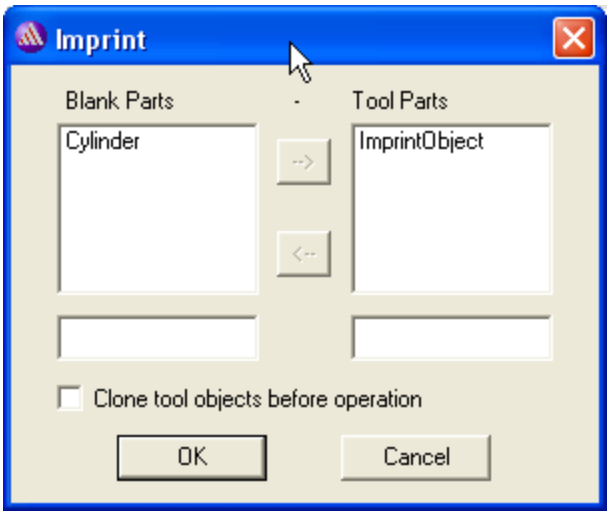
You can select the faces of the imprinted surface separately and assign properties as needed.



To imprint one object with another:

1. Select the intersecting objects.
2. Click **Modeler>Boolean>Imprint...**

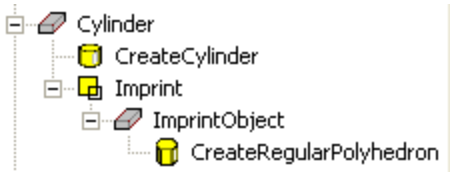
   This displays the Imprint dialog in which you designate which objects are the Blank Parts, and which the Tool Parts. If necessary, you can select the objects in lists, and use the arrow keys to move them. If desired, you can clone the tool objects before the imprint operation.



3. Click **OK**.

   This closes the dialog and performs the boolean imprinting.

After you perform the imprinting, the History tree retains the Imprint Object command and the create command for the imprinted object



If you select the Imprint command in the History tree, you can suppress the command via the Properties window. If you select the Create <*object*> icon for the object, you can edit the properties of that object. The changes applied to the object carry over to the imprinting.

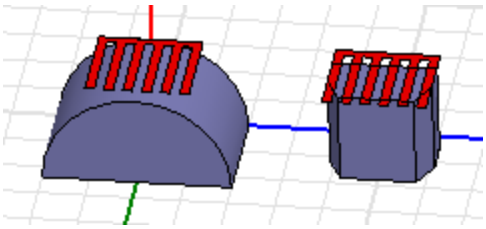
Related Topics

View and Edit Commands on History Tree Objects

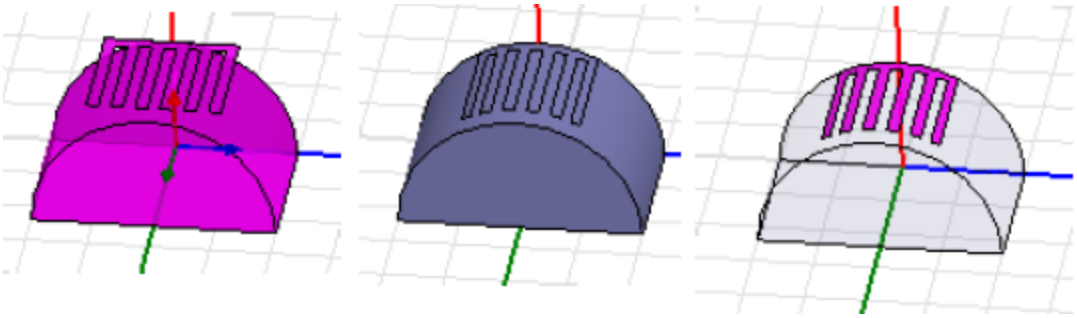
Imprint Projection Commands

Imprint Projection commands

The **Boolean>Imprint Projection** commands lets you project the form of one object to another surface. The receiving surface can be curved or faceted.



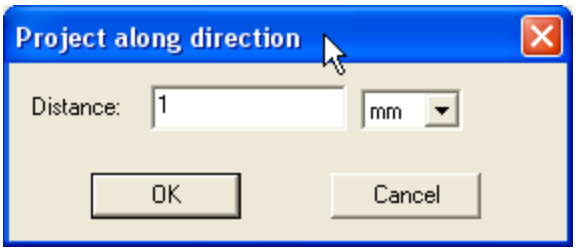
If the surface is curved, the dimensions of the projection will be affected.You can select the faces of the imprinted object separately, and edit properties as needed.



If projected shape extends beyond the face of the receiving object, the shape wraps.

1. Select the intersecting objects.
2. Click **Modeler>Boolean>Imprint Projection>Along Normal** or **Modeler>Boolean>Imprint Projection>Along Direction...**

   If you select **Along Normal**, the projection occurs along the normal. If you select Along Direction, you need to specify two points that describe the direction. Once you have defined a line by clicking two points, you see a dialog for specifying the distance for the projection.



3. Specify a distance and select units from the drop down menu and click OK.

   This closes the dialog and performs the boolean imprinting.

After you perform the imprinting, the History tree shows the Imprint Object command and the create command for the imprinted object

If you select the Imprint Projection command in the History tree, you can suppress the command via the Properties window. If you select the Create <object> icon for the object, you can edit the properties of that object. The changes applied to the object carry over to the imprinting.

Related Topics

View and Edit Commands on History Tree Objects
Imprinting an Object

Purging the History

Each object is a sequence of modeler-based operations. The history for each object is shown under its name in the model tree. You can use the Purge History command to remove the history of operations while not affecting the geometry itself. This is useful when you want to perform healing operations on the object. If there is an object for which you want to keep the history, you should make a copy of the object for that purpose before purging.

To purge the history:

1. Select the object.
2. Select **Modeler>Purge History**.

The history for the model is purged, and the context for the **Undo** and **Redo** commands is updated.

Related Topics

Working with the History Tree
Generating History to Reproduce Portions of a Model

Generating History to Reproduce Portions of a Model

If a polyline object (line, spline, or arc), circle, or ellipse is imported or history was previously purged, you can click on the polyline object and select **Generate History** to reproduce the individual line segments used to create the polyline in the model history tree.

To reproduce the line segments in the model history tree:

1. Select the polyline object.
2. Click **Modeler>Generate History**.

Related Topics

Purging the History
Draw Polyline

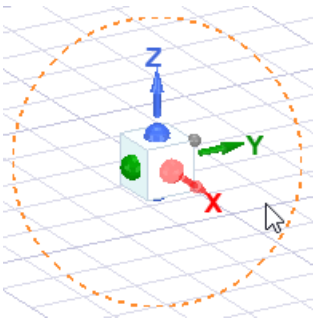
Modifying the 3D Model View

You can modify the view of contents in the **Modeler** window without changing their actual dimensions or positions within the coordinate system. You vary the viewing direction and the relative location of the "camera" (that is, the viewpoint) relative to the model geometry. The tools allow you to zoom, pan, or rotate the model viewpoint, and to quickly switch to pre-defined or user-defined viewpoints. Additionally, you can choose different object visualization options, show/hide objects, and customize the window background.

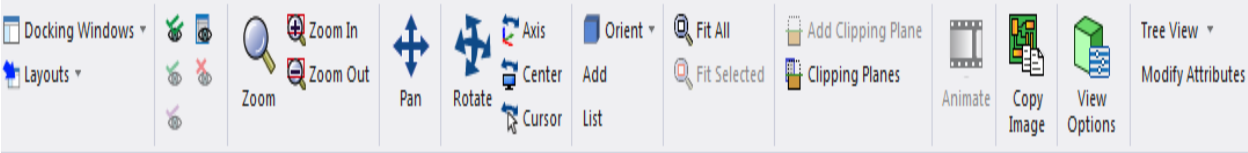
What do you want to do?

- Change the model view with Alt+double-click on areas in the modeler window.
- Set the View Options
- Rotate the view.
- Pan the view.
- Spin the view.
- Zoom in or out.
- Viewing 3D User Interface Options
- Fit contents in the view window.
- Show or Hideobjects.
- Render objects as wireframes, flat-shaded, or smooth-shaded solids.
- Set the Enhanced Display of Material Color and Transparency
- Modify the view orientation.
- Modify the lighting.
- Set the projection view.
- Set the background color.
- Modify the appearance of the coordinate system axes.
- Modify the appearance of the grid.
- Set the Surface Visualization
- Setting a symmetry multiplier
- Use Clip Planes

In addition to various menu bar and shortcut menu commands, you can manipulate the 3D model view using the Orientation Gadget:



Finally, you can also access many of the Modify View features via the **View Ribbon**. The following image has hot links for each area.

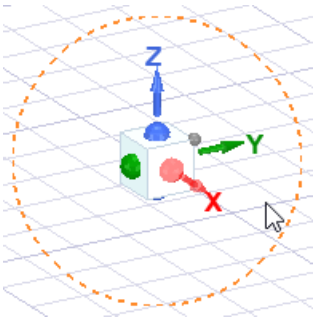


Related Topics

Assigning Color to an Object
Assigning Transparency to an Object

Using the Orientation Gadget

The *Orientation Gadget* is a tool for quickly manipulating the 3D model viewpoint graphically, rather than by navigating command menus. The gadget is enabled by default and appears near the lower left corner of the Modeler window's display area:



The orange dashed circle appears only when the cursor is near the gadget, and its purpose is discussed later in this topic.

There is a colored dot at the center of each face of the cube and red (x), green (y), and blue (z) miniaxes indicating the global axis positive directions. Additionally, there is a smaller gray dot at the corner where the +X, +Y, and +Z faces meet.

The following list summarizes the results of clicking different parts of the gadget (and optionally dragging the mouse while clicking). The descriptions are based on the behavior when the *Global* coordinate system is active:

- **Click the large dot at the center of any of the six cube faces:**

  This action rotates the viewpoint to place the side of the model that the clicked face represents towards you, parallel with the screen. For example, if you click the dot on the -Y face, the -Y side of the model faces you. Therefore, you would be looking in the +Y direction, which is the *Left* view.

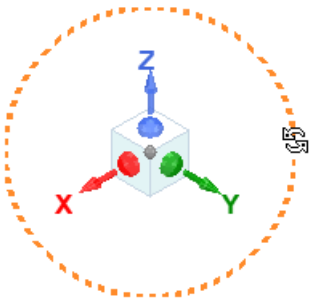
> **Note:** If the current viewing direction corresponds to one of the global axes (±X, ±Y, or ±Z), clicking the axis label or dot at the center of the orientation gadget (on the face that is towards you) reverses the model viewpoint. So, you can switch from front-to-back, right-to-left, top-to-bottom, or the inverses of these viewpoint changes with a single click of the mouse. This behavior holds true even for relative coordinate systems, as long as the current viewing direction corresponds to a global axis.

- **Click on one of the miniaxes:**

  This action is equivalent to clicking the dot at the center of the same positive cube face. The axis orientation becomes perpendicular to the screen and towards you. Therefore, the resulting three views are looking at the +X (front), +Y (right), or +Z (top) side of the model.
- **Click the small gray dot at the cube's +X, +Y, +Z corner:**

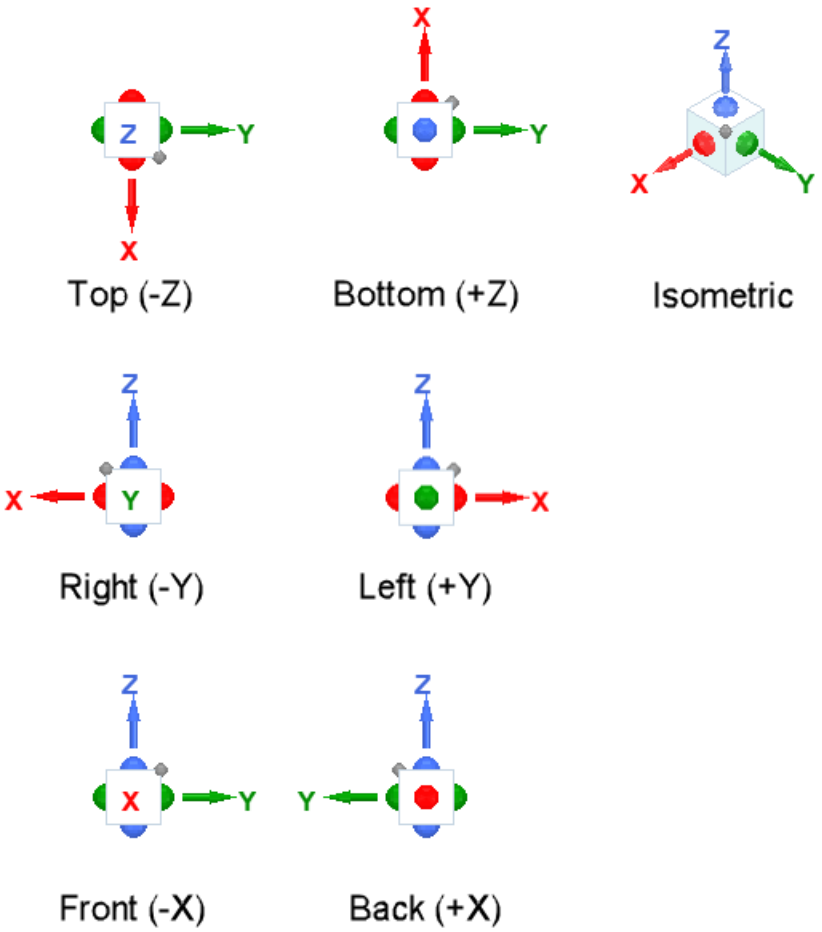
  This action produces the standard *Isometric* model viewpoint.
- **Click on or very close to the cube and drag the mouse:**

  This action freely rotates the model viewpoint. The rotation center depends on the **Default Rotation About** option selected in the *3D UI Options* dialog box. The behavior is equivalent to clicking and dragging in the display area using the middle button / wheel but is performed with the left mouse button.
- **Click on the orange dashed circle and drag the mouse:**



  This action rotates the model view about an axis normal to the screen. That is, the model view rotation is constrained to the plane of the screen regardless of the current viewing direction. The line weight of the dashed orange circle becomes heavier when the cursor is touching it, indicating a correct clicking point for constrained rotation.

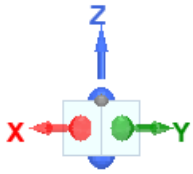
The appearance of the Orientation Gadget for each of the standard views is as shown in the following image:



Note:

The cube orientation, colored dots, and miniaxes of the Orientation Gadget always correspond to the Global coordinate system. However, the result of clicking on a dot or axis is a viewpoint that conforms to any currently active coordinate system. See the example that follows.

Example: Assume that you have defined a relative coordinate system in which the Ux axis is rotated counter-clockwise 45° from the Global X axis, in the Global XY plane, and that this relative coordinate system is active. If you click on the X miniaxis of the Orientation Gadget, the model viewpoint will place the Ux axis perpendicular to the screen. The appearance of the gadget after clicking would be as shown below:

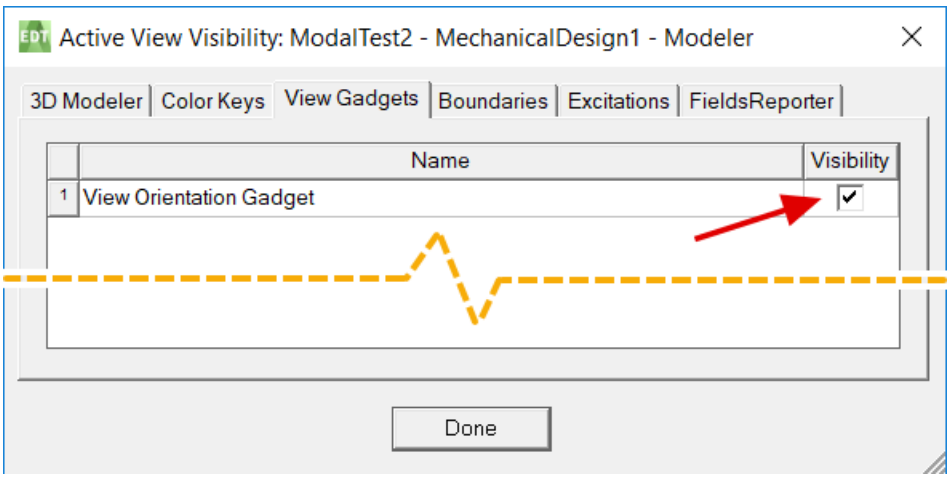


For the same relative coordinate system example, clicking the Z axis of the Orientation Gadget will produce the following result:

Controlling Visibility of the Orientation Gadget

You can enable or disable the Orientation Gadget from the, as follows:

1. Access the *Active View Visibility* dialog box using one of the following two methods:
   - From the menu bar, click **View> Visibility> Active View Visibility**.
   - On the **View, Draw,** or **Model** ribbon tab, click **Hide/Show overlaid visualization in the active view"**.
2. Select the **View Gadgets** tab.
3. Select or clear the checkbox in the **Visibility** column to the right of **View Orientation Gadget** to control the gadget visibility.



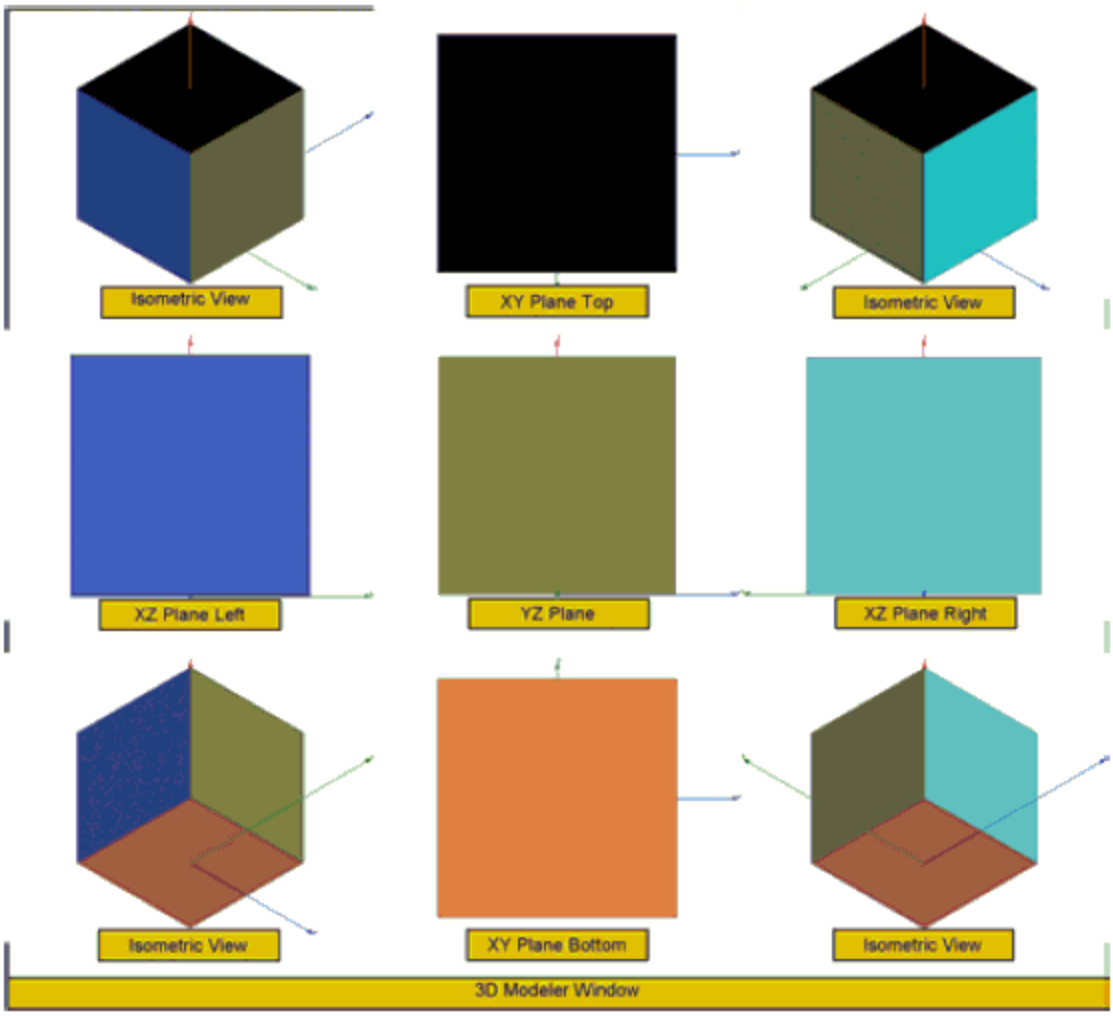
4. Click **Done** to close the dialog box.

Related Topics:

Modifying the Model View

Changing the Model View with Alt+Double Click Areas

In the following figure, the orientation of the colored boxes shows the nine model orientations you can obtain by using **Alt + double click** in the corresponding area of the Modeler window.

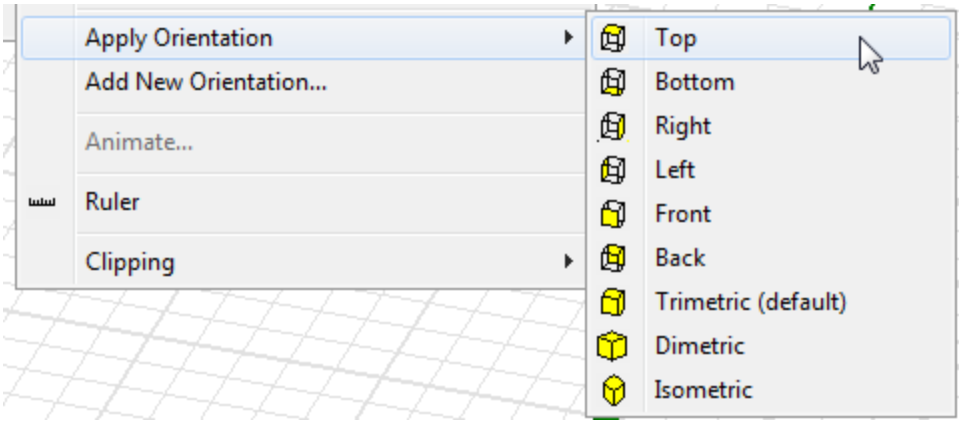


Related Topics

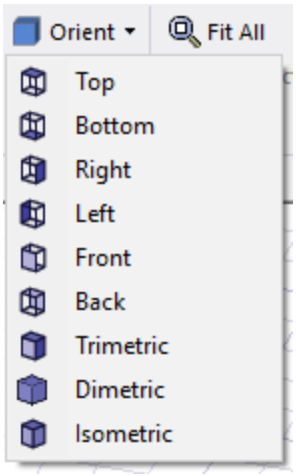
Rotate the View.

Apply an Orientation to the Current View

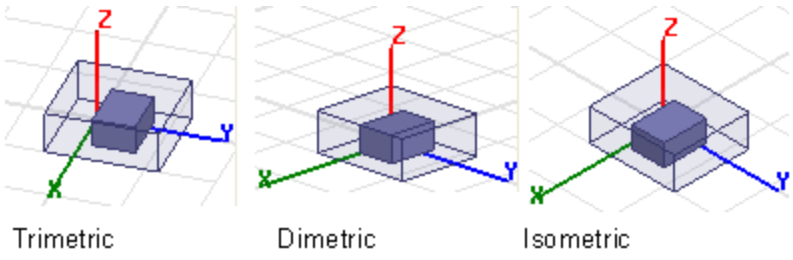
Right-click in the Modeler window to see the **View** shortcut menu with the **Apply Orientation** command.



You can also use the equivalent View ribbon icons.

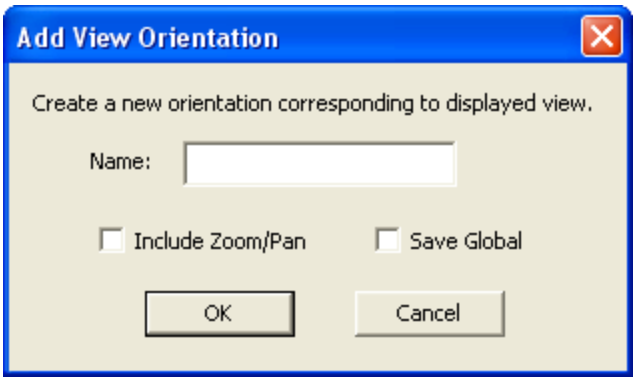


Possible orientations are Top, Bottom, Right, Left, Front, Back, Trimetric, Dimetric, and Isometric. These last three differ as shown below.



After applying an orientation, you can **View>Undo View:Orientation**.

You can also use the right-click **View** shortcut menu to **Add New Orientation**.



The Name that you assign will appear on the Apply Orientation command menu. You can specify whether to Include Zoom/Pan and whether to Save Global.

Related Topics

Change the Model View with Alt-Double Click Areas

Viewing 3D User Interface Options

Use the **View>Options** command to open the **3D UI Options** dialog box. This lets you set defaults for the following view options:

- **Stereo mode(Default is disabled)**

  This mode is designed for stereo shutter glasses. It should remain disabled when not in use as it causes blurring of the screen. This mode requires special graphics hardware and drivers for OpenGL page flipping. If you hardware does not support this mode, you will receive an error.
- **Drag Optimization(Default is disabled)**
- **Show Ansoft logo in prints** (Default is disabled)
- **Default Color Key height** - the maximum number of values displayed (Default is 16)
- Under **When there is a selection**, you may set the following:
  - **Selection always visible** (Default is enabled)
  - **Set transparency of selected objects** (Default is enabled with value set at 0.1)
  - **Set transparency of non-selected objects** (Default is enabled with value set at 0.9)
- Under **Default Rotation About**, you may select one of the following:
  - **Screen Center (Default)**
  - **Current Axis**
  - **Model Center**
  - **Cursor**

Rotating the View

You can rotate the view relative to the Model Center, the Screen Center, or the Current Axis.

When you select one of these as the View>Options default, the **View>Rotate** selection menu changes to show that the Alt-Drag combination attaches to your selection.

To rotate the model center from the menus or icons:

1. Click **View**>**Interaction>Rotate Model Center** or click the **Rotate** icon on the **View** ribbon.
2. Drag the mouse in the direction you want to rotate the view.

   The view rotates until you release the mouse button.
3. To exit **Rotate** mode, click **View>Interaction>Rotate Model Center** again, or click the icon again or press **ESC**.

Hint	Alternatively, rotate the view using one of the following methods: • Hold down the **ALT** key as you drag the mouse. • Right-click in the view window, and then click **View>Rotate** on the shortcut menu.

To rotate the view around the current axis:

1. Click **View**>**Interaction>Rotate Current Axis**.
   - Alternatively, click **View>Interaction>Spin**.
2. Drag the mouse left or right at the speed you want to spin the view.

   The view spins continually in the direction and at the speed you dragged the mouse.
3. To stop spinning the view, click in the view window.
4. To end **Spin** mode, click **Spin** again on the **View>Interaction** menu or press **ESC**.

To rotate the view around the screen center:

1. Click **View>Interaction>Rotate Screen Center** or click the rotate around screen **Center** icon on the Model ribbon.
2. Drag the mouse up and down at the speed you want to rotate the view.
3. To end **Rotate** mode, click **View>Interaction>Rotate Screen Center**, or click the icon again or press **ESC**.

Related Topics

Pan the view

Panning the View

To move (pan) the view:

1. Click **View**>**Pan**  .
2. Drag the mouse in the direction you want to pan the view.

   The view will pan until you release the mouse button.

3. To exit **Pan** mode, click **Pan** on the **View** menu again or press **ESC**.

Hint	Alternatively, pan the view using one of the following methods: • Hold down the **SHIFT** key as you drag the mouse. • Right-click in the view window, and then click **View>Pan** on the shortcut menu.

Spinning the View

1. Click **View>Spin**.
2. Click-and-drag the mouse in the direction and at the speed you want to spin the view.
   The view spins continually in the direction and at the speed you dragged the mouse.
3. To stop spinning the view, click in the view window.
4. To end **Spin** mode, click **View>Spin** again, or press **Esc**.

Hint	Alternatively, right-click in the view window, and then click **View>Spin** on the shortcut menu.

Zooming In and Out

You can magnify (zoom in) or shrink (zoom out) the contents in the view window using hot keys, or mouse drag zoom mode, use **Zoom In** and **Zoom Out** commands for 5% increments, and use **Zoom Previous** to undo recent commands, as described below. You can also Zoom out on a Rectangular Area, Zoom in or Out Using a Mousewheel, or Zoom to Selected Excitation.

> **Note:** Some international keyboards may not support the following hotkeys. If the Shift key is required for a character, such as the plus sign (+), the hotkey may not function.

To zoom in using hotkeys:

- Press the plus sign (**+**) or equals sign (**=**) key
- Press **Ctrl++**

  The view zooms in 5 percent.

To zoom out using hotkeys:

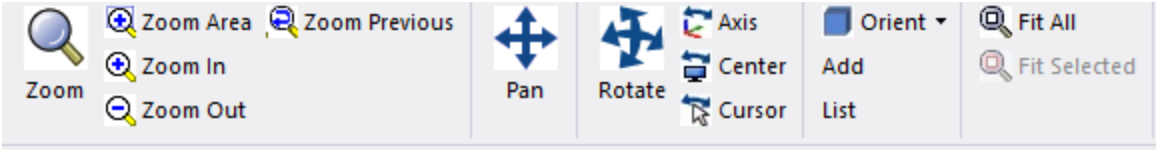
- Press the minus sign (**-**) key
- Press **Ctrl+-**

  The view zooms out 5 percent.

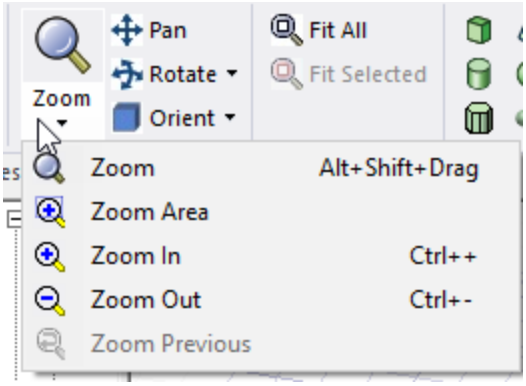
To zoom by dragging the mouse.

1. Click **View** > **Interaction** > **Zoom**, click **View** > **Zoom** on the shortcut menu, or click the **Zoom** icon on the **View, Draw,** or **Model** ribbon tabs.

   View ribbon tab with **Zoom**.



Draw or **Model** ribbon tabs with Zoom icon, and drop-down menu for Zoom:



2. To zoom in, drag the mouse toward the top of the view window. The objects in view expand as you drag.

   To zoom out, drag the mouse toward the bottom of the view window. The objects in view decrease in size as you drag.

   When zooming on a view of model objects the absolute size of the model does not change.

   When zooming on a 2D report, axis labels and ticks will adjust automatically during the zoom operation and will rescale to their final value after the zoom operation is complete.
3. To end **Zoom** mode, click **View** > **Interaction** > **Zoom** again, or click the Zoom icon, or press **Esc**.

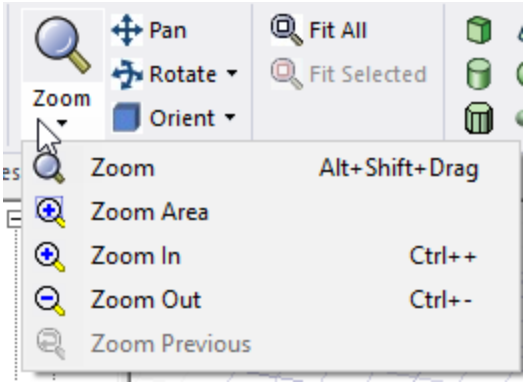
Zoom In and Zoom Out Commands

To use **Zoom In**, **Zoom Out** to zoom in or out in 5% increments:

1. Click **View** > **Interaction** > **Zoom**, click **View** > **Zoom In** or **Zoom Out** on the shortcut menu, or click the **Zoom** icon on the **View, Draw,** or **Model** ribbon tabs.

   View ribbon tab with **Zoom In** and **Zoom Out**.

Draw or **Model** ribbon tabs with **Zoom** icon, and drop-down menu for **Zoom In** and **Zoom Out**:



2. Click **Zoom In** or **Zoom Out** for a 5% increment zoom in or out respectively.

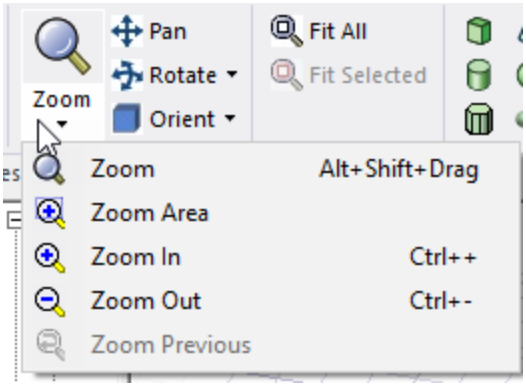
   To zoom out, drag the mouse toward the bottom of the view window. The objects in view decrease in size as you drag.

   When zooming on a view of model objects the absolute size of the model does not change.

   When zooming on a 2D report, axis labels and ticks will adjust automatically during the zoom operation and will rescale to their final value after the zoom operation is complete.
3. To end **Zoom** mode, click **View** > **Interaction** > **Zoom** again, or click the **Zoom** icon, or press **Esc**.

Zoom Previous Command

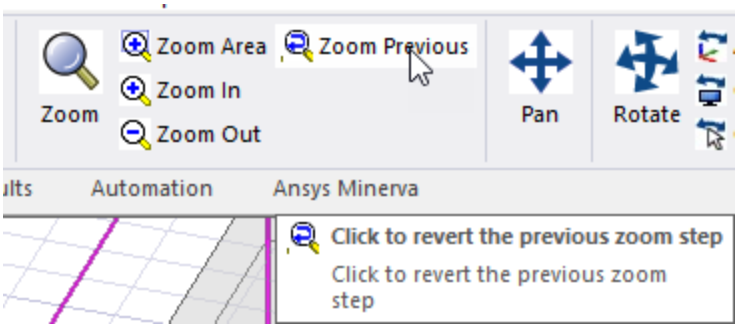
The **Zoom Previous** command is not enabled until at least one **Zoom** command has been used.



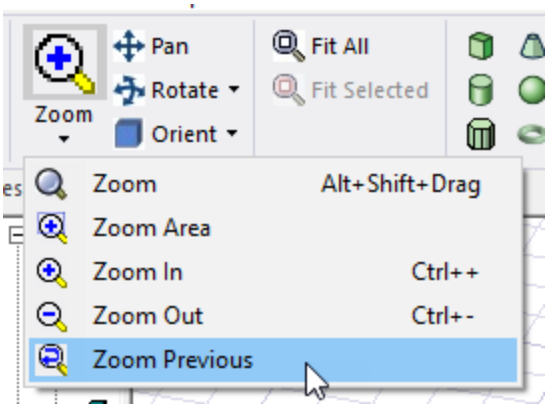
It operates on any existing sequence of **Zoom** commands. That is, it undoes the last previous **Zoom** command, and then, another exists, the one before that. To use **Zoom Previous**.

1. Click **View** > **Interaction** > **Zoom Previous**, click **View** > **Zoom Previous** or Zoom Out on the shortcut menu, or click the **Zoom** icon on the **View, Draw,** or **Model** ribbon tabs.

   View ribbon tab with .



Draw or **Model** ribbon tabs with **Zoom** icon, and drop-down menu for **Zoom Previous** :



2. Zoom Previous undoes the last executed Zoom command, and then, if others exist, can be used to undo them in reversed order of execution.

   When zooming on a view of model objects the absolute size of the model does not change.

   When zooming on a 2D report, axis labels and ticks will adjust automatically during the zoom operation and will rescale to their final value after the zoom operation is complete.
3. To end **Zoom** mode, click **View** > **Interaction** > **Zoom** again, or click the **Zoom** icon, or press **Esc**.

Tip:

Alternatively, zoom in or out using one of the following methods:

- Hold down the **Alt**+**Shift** keys as you click and drag the mouse.
- Right-click in the Modeler window and then click **View** > **Zoom** on the shortcut menu.

Note:

If Zooming is slow, especially for complex models, for some graphics cards, you can improve performance by setting **NVIDIA Control Panel** > **3D Settings** > **Manage 3D Settings** > **Global Settings** > **Global Presets: Workstation App - Dynamic Streaming**

For more information about graphics card see the section *Open GL* under *Installation Prerequisites* in the *Ansys EM Installation Guide*.

Related Topics

Zooming In or Out on a Rectangular Area
Zooming In or Out using a Mousewheel
Fitting Objects in the View Window
Zoom to Selected Excitation

Zooming In or Out on a Rectangular Area

To magnify or shrink a specific rectangular area in the view window:

1. Click **View**>**Zoom In**  or **View>Zoom Out**  .
   - Alternatively, right-click in the view window, and then click **View>ZoomIn** or **View>Zoom Out** on the shortcut menu.
2. Use the mouse to draw a rectangle (or square) by selecting two diagonally opposite corners.

   This is the area of magnification to be increased or decreased.

   The rectangular area is magnified or decreases in size.

   When zooming on a view of model objects, the absolute size of the model does not change.

   When zooming on a 2D report, axis labels and ticks will adjust after the zoom operation is complete.
3. To end **Zoom** mode, click **Zoom In** or **Zoom Out** on the **View** menu again, or press **Esc**.

Related Topics

Zooming In and Out
Fitting Objects in the View Window

Zooming In or Out using a Mousewheel

By clicking in the Modeler window you can use the mousewheel to zoom in and out of the 3D Modeler window. The cursor location becomes the center relative to the zoom, rolling the wheel

forward increases the zoom, and rolling the wheel backwards decreases the zoom. If the window includes a ruler display, this adjusts as you scroll in and out.

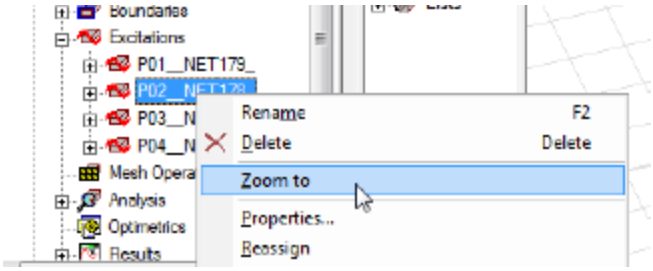
> **Note:**
>
> If Zooming is slow, especially for complex models, for some graphics cards, you can improve performance by setting **NVIDIA Control Panel** > **3D Settings** > **Manage 3D Settings Global Settings** > **Global Presets: Workstation App - Dynamic Streaming**
>
> For more information about graphics card see the section **Open GL** under **Installation Prerequisites** in the **Ansys EM Installation Guide**.

Zoom to Selected Excitation

Zoom to is an option that can enlarge the view of an excitation. This can be very useful if you want to inspect any problem areas.

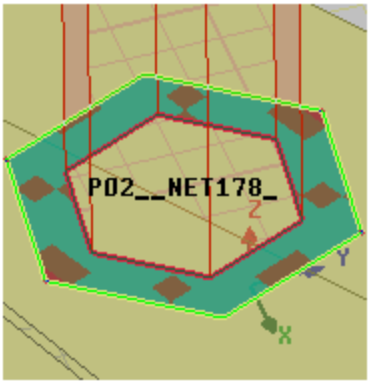
You can use the **Zoom to** option as follows.

1. Select the port under **Excitations** in the project tree to highlight it in the geometry.
2. Right-click to generate the shortcut menu as shown below.



3. Select the option **Zoom to**.

   Notice that the option instantly increases magnification of the surface excited with the wave port as shown in the Figure below.



Related Topics:

Modifying the Model View

Fitting Objects in the View Window

What do you want to do?

- Fit all objects or traces in a view window.
- Fit selected objects in a view window.

Fitting All Objects in a View Window

To fit all the views: click **View>Fit All>All Views**.

All view windows displaying the active design change to include all model objects.

To fit only the active view: click **View>Fit All>Active View**.

The view in the active **Modeler** window changes to include all model objects.

Hint	Alternatively, fit all objects in the *active* view window using one of the following methods: • Press **CTRL+D**. • Right-click in the view window, and then click **View>Fit All** on the shortcut menu.

When **Fit All** is used in a **Report** view, the window is automatically rescaled to fit all traces in the window and the axis label and ticks are rescaled.

Related Topics

Fitting a Selection in a View Window

Fitting Selected Objects in a View Window

To fit one or more selected objects, 3D Components, or User Defined Models in the Modeler window or to fit traces selected in the Reporter:

1. When you are working on a model view, select the objects, 3D Components, or User Defined Models you want to fit in the view. When you are working on a report, select the traces you want to fit.
   - To fit the selection in the active view window: **Click View>Fit Selection>Active View**.
   - To fit the selection in every open view window of the active design: **Click View>Fit Selection>All Views.**

Hint	Alternatively, fit the selection in the active view window by clicking **View>Fit Selection** on the shortcut menu.

To fit one or more objects or components selected in the History tree:

1. Select the objects or components of interest, and right click to display the menu.
2. Select **View>Fit in Active View** or **Fit in All Views**:

   The view adjusts to fit the select objects.

Related Topics

Fitting All Objects in a View Window

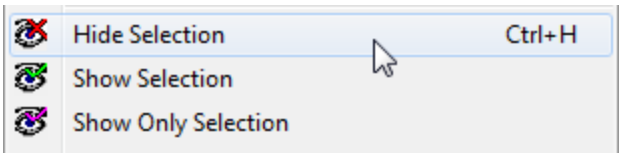
Showing Only Selected Objects in All or Active Views

Hiding Objects from View

To hide selected objects.

1. Select the object you want to hide from view. This enables the menu commands and the **View**, **Draw**, and **Model** ribbon tabs icons for hiding objects.
2. Click **View>Visibility>Hide Selection** and one of the following commands:
   - **All Views** to hide the selected object in every open view window.
   - **Active View** to hide the selected object in the active view window. You can also use CTRL-H to hide a selected object.

   You can also right-click in the modeler window for a short-cut menu and use the **View** commands for **Hide Selection**, or **Show Only Selection**.



You can also use the **Hide** selected objects icons on the **View**, **Draw**, and **Model** ribbon tabs to hide selected objects in all views or the active view.

The objects you selected are hidden.

If there are many objects, you may find it easier to hide objects using the **Active View Visibility**dialog box. You can also chose to show only selected objects in all or active views, effectively hiding all unselected objects.

Note	Hiding boundaries also turns off a check for boundary overlaps during boundary assignment. In the case of very large models with many boundaries, hiding boundaries can prevent delays during boundary assignment. Full model validation will subsequently check for boundary overlaps.

Object visibility is saved with the project.

Related Topics

Showing Objects

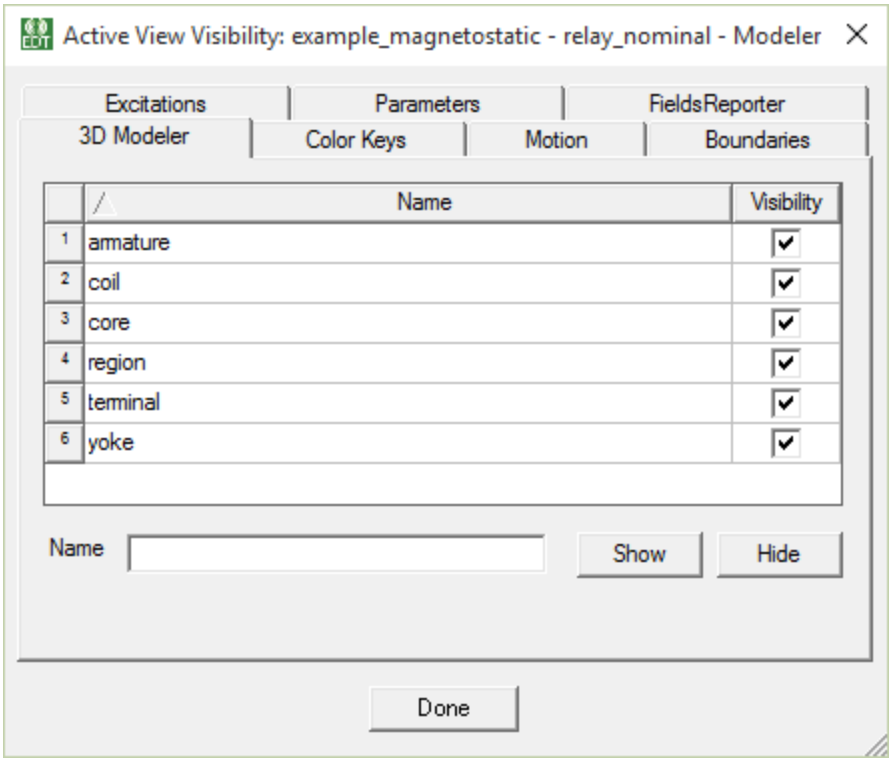
Showing Only Selected Objects in All or Active Views

Active View Visibility

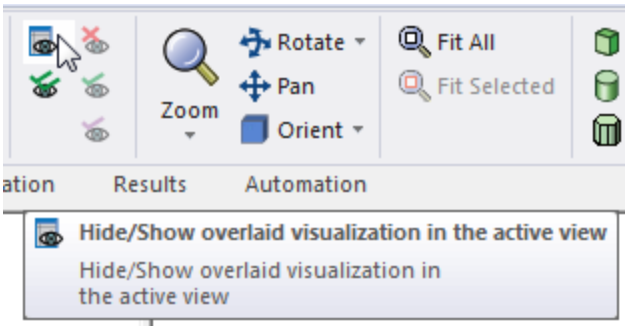
Showing Objects

You can show or hide objects in several ways, using the **Active View Visibility** dialog for individual objects, or **Visibility** menu selections for All Objects, Only Selection, Model Objects, Non-model Objects, Primitive Shapes, or Non-primitive shapes.

The **Active View Visibility** dialog contains tabs for 3D Modeler, Color Keys, Array Setup, Boundaries, Excitations, and Fields Reporter. Select the tab for the objects you want to show or hide, and check or uncheck **Visibility**.



You can access the dialog from the **View>Visibility>Active View Visibility...** menu item or by clicking the **Hide/Show** visualization icon on the ribbon **View**, **Draw**, and **Model** tabs.



The **Visibility** menus are available on the **View** menu and as short-cut menus in the modeler window and in the history tree when you select the Model icon

To show *one or more objects* that are currently hidden:

1. Click **View>Visibility>Active View Visibility** , or click the ribbon visibility icon  .
   The **Active View Visibility**dialog box appears.
2. Select the tab for the objects you want to show or hide. The dialog contains tabs for:
   - 3D Modeler Objects
   - ColorKeys

- MotionSetup Indicators
- Boundary Indicators
- Excitations Indicators
- Parameter Indicators
- FieldsReporter

3. Under the tab you need, select the **Visibility** option for the objects you want to show in the active view window.
   - For designs with large numbers of objects, you can resize the dialog for easier selection.
   - By default, objects are listed in alphabetical order. You can invert the order by clicking the Name bar above the Name fields. A triangle in the bar indicates the direction of the listing.
   - You can also use the **Name** field to type in an object name and apply the visibility via the **Show** and **Hide** buttons.

   The objects you select and designate as Visible (by selecting the property or using Show) reappear.

To show *all objects* that are currently hidden:

1. Click **View>Visibility>Show All** and one of the following commands:
   - **All Views** to show all objects in every open view window
   - **Active Views** to show all objects in the active view window.

   The selected objects reappear.

To show *selected objects* that are currently hidden:

1. Select the object. Hidden items are selected once the node corresponding to them is clicked in the history tree.
2. Click **View**>**Show Selection** and one of the following.
   - **All Views** to show selected objects in every open view window
   - **Active Views** to show selected objects in the active view window.

   You can also use the ribbon **View** tab icons to **Show selected objects in active views** or open the **Active View Visibility** dialog..

To show one or more *3D components* that are currently hidden:

1. In the history tree, select one or more 3D components. This activates the **View**, **Draw**, and **Model** tab **Hide/Show** icons in the ribbon and the **View>Visibility** menu selections.
2. You can right click to display the shortcut menu and select **View>Visibility>Show All**.

Object visibility is saved with the project.

Related Topics

Hiding Objects

Showing Only Selected Objects in All or Active Views

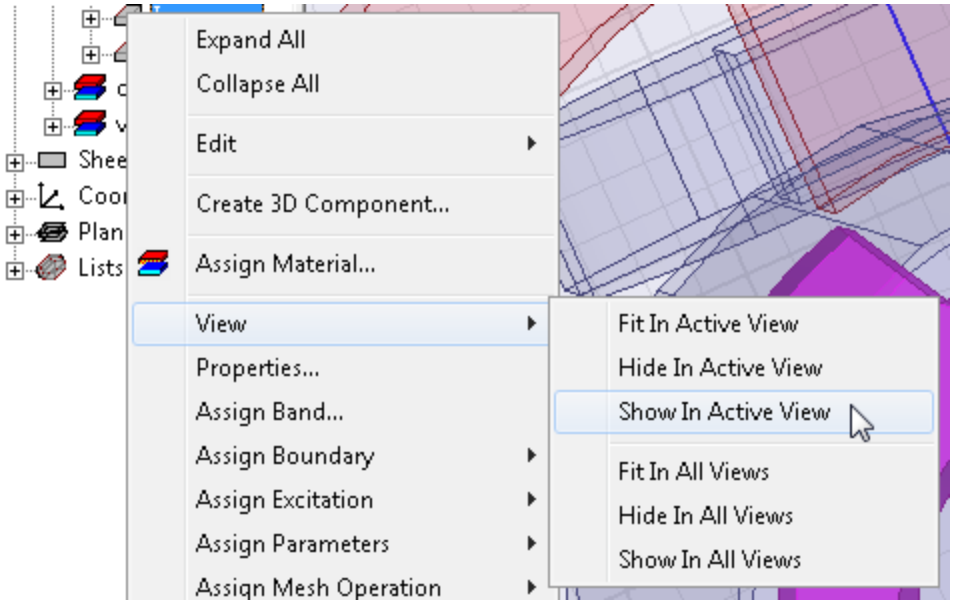
Active View Visibility

Showing Only Selected Objects in All or Active Views

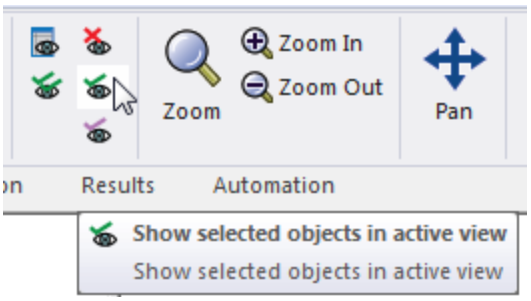
To show only *selected objects*:

1. Select the object from the Project tree or history tree. Hidden items are selected once the node corresponding to them is clicked in the history tree.
2. Click **View>Visibility>Show Selection** and one of the following.
   - **All Views** to show selected objects in every open view window
   - **Active Views** to show selected objects in the active view window.

   You can also right click, and use the shortcut menu:
3. You can right click to display the shortcut menu and select **View>Show in Active View** or **Show in All Views.**



You can also use the ribbon View, Draw, and Model tab **Show selected objects in all views** and **Show selected objects in active views** icons.



The selected objects reappear.

Active View Visibility Dialog

If there are many objects, it may be easier to show or hide objects using the ***Active View Visibility*** dialog

1. Click **View**>**Visibility>Active View Visibility**, or click the **Hide/Show** icon on the **View**, **Draw**, or **Model** ribbon tabs.

   The **Active View Visibility** dialog box appears.
2. Select the tab for the objects you want to show or hide. The dialog contains tabs for 3D Modeler objects, Color Key objects, Boundaries, Excitations, and Fields Reporter objects.
   - For designs with large numbers of objects, you can resize the dialog for easier selection.
   - By default, objects are listed in alphabetical order. You can invert the order by clicking the Name bar above the Name fields. A triangle in the bar indicates the direction of the listing.
   - You can also use the **Name** field to type in an object name and apply the visibility via the **Show** and **Hide** buttons.
3. Under the tab, clear the **Visibility** option for the objects you want to hide in the active view window.

   The objects you designate are hidden.

Note	Hiding boundaries also turns off a check for boundary overlaps during boundary assignment. In the case of very large models with many boundaries, hiding boundaries can prevent delays during boundary assignment. Full model validation will subsequently check for boundary overlaps.

Object visibility is saved with the project.

Related Topics

Showing Objects

Showing Only Selected Objects in All or Active Views

Hiding Objects

Rendering Objects as Wire Frames or Solids

To render (display) *all* objects in the view window as wireframe outlines or smooth-shaded solids:

1. Click **View**>**Render**.
2. On the **Render** menu, click one of the following:
   - **Wire Frame**. The objects in the view window are displayed as skeletal structures, enabling you to see all sides of the objects at one time.

     You can also use the **F6** key or the shade icon  to toggle the display to wireframe.
   - **Smooth Shaded**. The objects in the view window are displayed as shaded objects with smooth edges.

You can also use the **F7** key or the shade icon  to toggle the display to smooth shaded.

To render *a single* object in the view window as a *wireframe outline*:

1. Select the object you want to render as a wireframe:
2. In the Properties dialog box, under the **Attribute** tab, select the check box for the **Display Wireframe** property.

Hint	You can also press **F6** (Wire Frame) and **F7** (Smooth Shaded) to toggle between these two views.

Related Topics

Setting the Default Rendering Mode

Setting the Curved Surface Visualization

Setting the Default Rendering Mode

To set a default rendering mode for all objects created in the active design and in future designs:

1. Click **Tools>Options>General Options** to open the **Options** dialog displaying a hierarchical list of options.
2. Click the + next to **3D Modeler** to expand the list.
3. Click the + next to **Display**, and select the **Rendering** option to display the various rendering options .
4. Select one of the following from the **Default view render mode** pull-down list.
   - **WireFrame**. The objects in the view window are displayed as skeletal structures, enabling you to see all sides of the objects at one time.
   - **SmoothShade**. The objects in the view window are displayed as shaded objects with smooth edges.
4. Click **OK**.

The rendering mode is applied to all new objects you create.

Related Topics

3D Modeler Options: Display

Setting the Enhanced Display of Material Color and Transparency

You can specify the use of material color and transparency display by using the **View>Render>Enhanced Display** command. Any changes reset the default.

To set the Enhanced Display for the active modeler window:

1. Click **View>Render>Enhanced Display** or press F8.

Enhanced Display overrides a part's color and transparency properties with material settings for all the parts visible in the active view. By default it is off (unchecked on the **View>Render** menu) and the part's attribute color and transparent values are applied.

It is possible for the view to be **Enhanced Display** mode, yet some of the parts in model could have **Use Material Appearance** unchecked. In this case you can edit color or transparent values or **Use Material Appearance** in the Property window but they are applied to active view visualization.

Sheet objects in the 3D modeler do not have a material assignment, but they could have a boundary assignment with material assigned to the boundary. In this case sheet object visualization resembles solid object visualization where material appearance from the assignment is used.

Related Topics

Rendering Objects as Wireframes or Solids

Setting the Curved Surface Visualization

Maxwell allows you to specify the faceting for rendering true curves by using the **View>Render>Curved Object Visualization** command. There are two options for control--Maximum surface deviation and Maximum normal deviation. This resembles the Mesh surface approximation settings. Reduce either or both of the allowed deviations to improve the image quality. Improved image quality comes at the cost of increased CPU consumption. Changes apply to the current model until they are changed again. Any changes reset the default.

Legacy projects with non default settings are approximately converted to new settings with warning. Overall faceting time may be around 20% more but number of triangles should reduce.

The default gives satisfactory results (i.e. cpu/memory consumption vs. graphical display) for various model complexities.

When you change Curved Object Visualization settings and apply them to a design, those settings are saved with design unless you change it again. That means when you open the design again, it will apply saved visualization settings and NOT the default settings. Because this affects the CPU and memory required to open the project, typically, you should not save a project with other than the default settings.

To set the Curved Object Visualization settings for the active modeler window:

1. Click **View>Render>Curved Object Visualization**.

   This command displays the **Curved Object Visualization** dialog for the active modeler window. The dialog contains areas for setting the Maximum Deviation, and the Maximum Normal deviation.

2. Set the Maximum Deviation by first selecting from the radio buttons for **Ignore**, set as **Relative Deviation** or set as **Absolute Deviation**. Selecting the later two radio buttons enables the value field.

   When set as **Relative Deviation**, the actual surface deviation depends on the model size. For example, sphere with a radius of 10 has same number of facets as a sphere with a radius of 1. This means that CPU cost does not increase based on the model dimension.

When set as **Absolute Deviation**, the maximum surface deviation for both the spheres will be approximately same since a bigger sphere has more facets than a smaller one. This means that the most CPU cost applies to the larger objects.

3. If you selected the radio buttons for Relative or Absolute Deviation for Maximum Deviation, enter a value in the field.
4. To change the **Maximum Normal Deviation**, enter a value in the text field. Units are degrees.

Note	Wire bodies cannot be rendered with a Maximum Normal Deviation value less than 1 degree. When using a setting less than 1 degree all wire bodies will be rendered with a setting of 1 degree and all closed bodies will be rendered with the dialog box setting.

5. The **Save As Default** button lets you Save any values you change to the drop down menus for the fields.
6. The **Restore Default**s button lets you return to the original values. Any values you provided through **Save As Default** remain on the drop down menus for the fields for surface and normal deviations
7. Click **Apply** to apply the current values to the active modeler window, and **Close** or **Cancel** to close the dialog without changing settings.

Related Topics

Rendering Objects as Wireframes or Solids

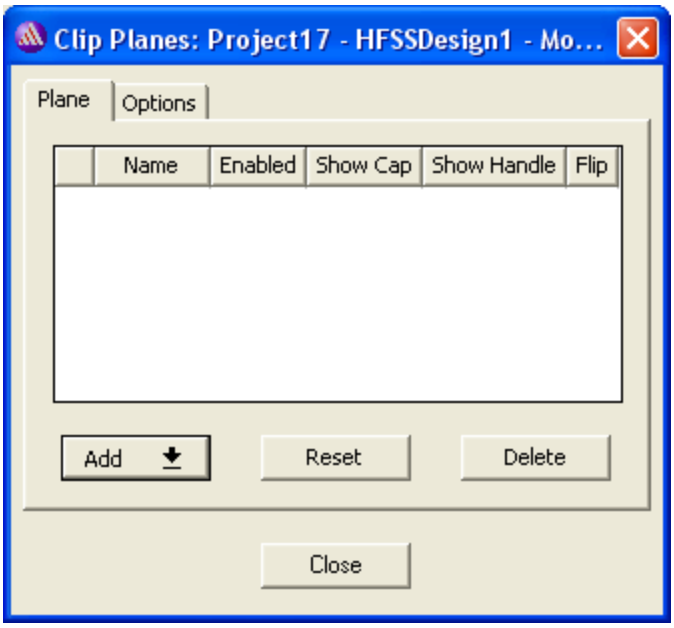
Using Clip Planes

The **View>Clip Plane** command lets you define a clip plane that you can use to interactively make any desired cut-away view of a model. If you use **Edit>Copy Image** or **Modeler>Export>***image format* with the clip plane active, the image shows the clipped plane. When parts of the model are hidden by a clip plane, model selection works as though only the visible parts are present.

To add a clip plane:

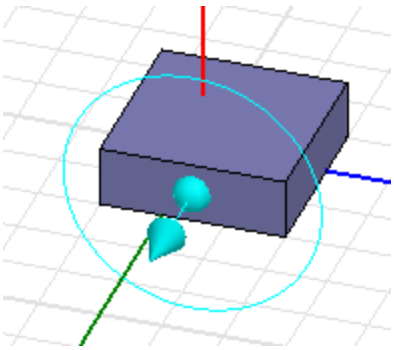
1. Click **View>Clip Plane**...

   This displays the **Clip Plane** dialog with the Plane tab selected.

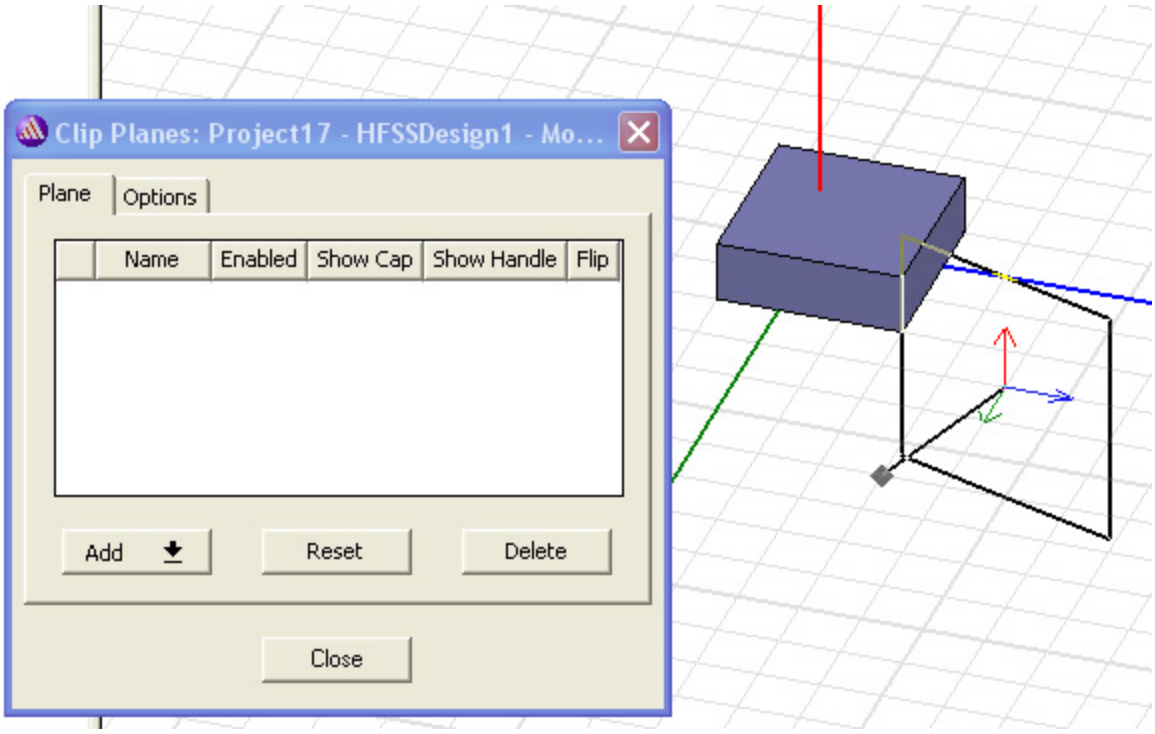


2. The **Add** button contains a drop down menu with choices for **Specify center, normal**, and **Use selection**. Of you want to use selection, you must first select a face or a cut plane.

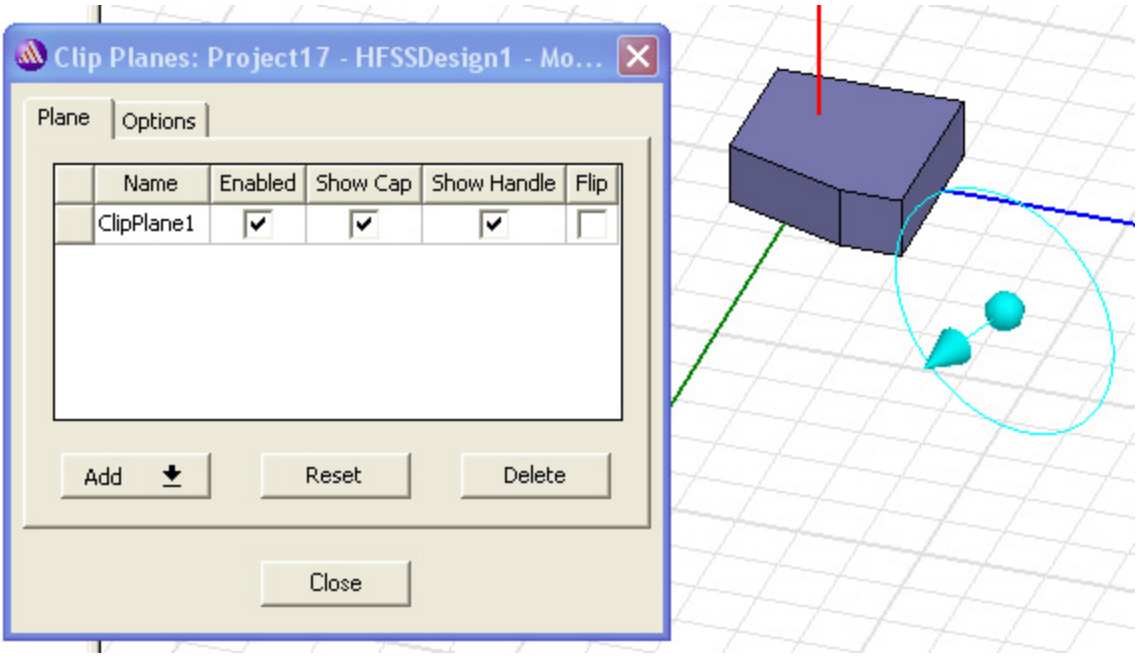
- If you first select a face or cut plane, and then click **Add>Use Selection**, the clip plane is added on that face.



a. If you select **Add>Specify center, normal**, this launches a Measure dialog and enters a mode for you click to first define the start location, shown as a triad. When you move the cursor, a rectangle represents the clip plane, and a vector the current direction.

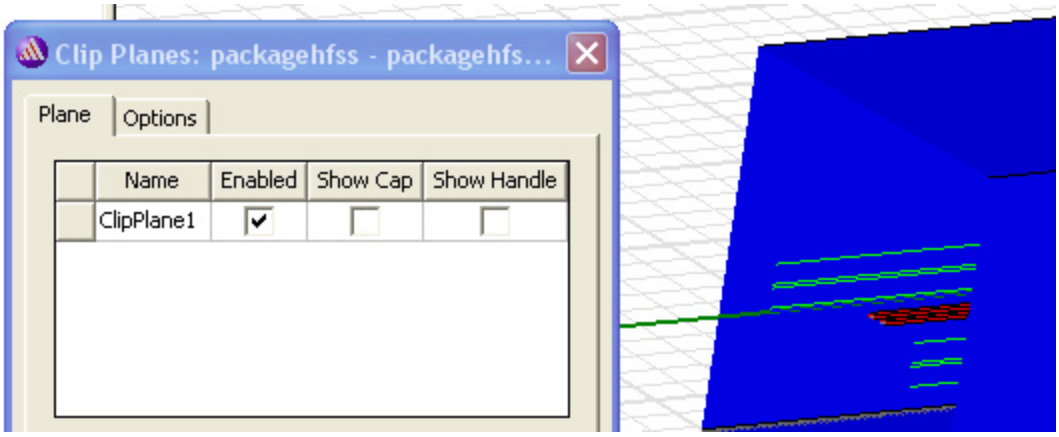


b. Click again to set the reference position.

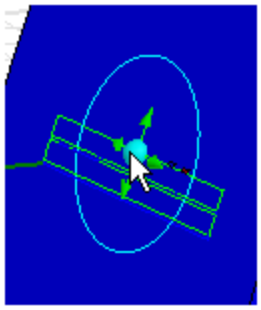


After the second click, the clip plane is active. The handle is visible as a circle with a sphere at the center, and an arrow pointing the normal for the plane. The **Clip Planes** dialog shows the clip plane name, that it is enabled, shows the cap (which is the plane

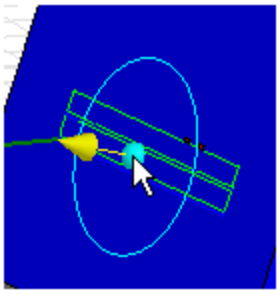
surface), and shows the handle.The Flip selection lets you reverse the direction of the clip plane. If you uncheck **Show cap** and/or **Show Handle**, they disappear from the display.



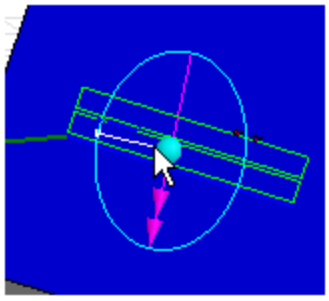
3. With Show Handle enabled, you can use the handle to manipulate the location and orientation of the clip plane. The handle changes appearance and function relative to the position of the cursor. Dragging the cursor makes use of the current function.



Move Handle independently of clip plane. This lets you move the handle away from the model.



Move Handle while dragging clip plane



Rotate clip plane by dragging the cursor. The white line is parallel to the clip plane rotation. The double arrow line is the axis.

Move the cursor around the center of the handle to change the axis.

4. The **Options** tab for the **Clip Planes** dialog contains four options.
   - Force opaque for the unclipped portion.
   - Disable clip plane when drawing a new clip plane.
   - Plane handle color

     The button shows the current color. Click the button to display a color selection dialog box. Select a default or custom color and click **OK**.
   - Plane handle radius.

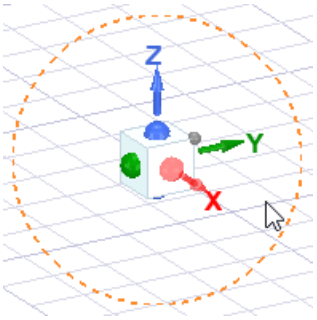
This slider lets you resize the radius of the handle to the most convenient size. The radius resizes dynamically. When you close and reopen the modeler window, the last selected size persists.

You can save your choices as new defaults.

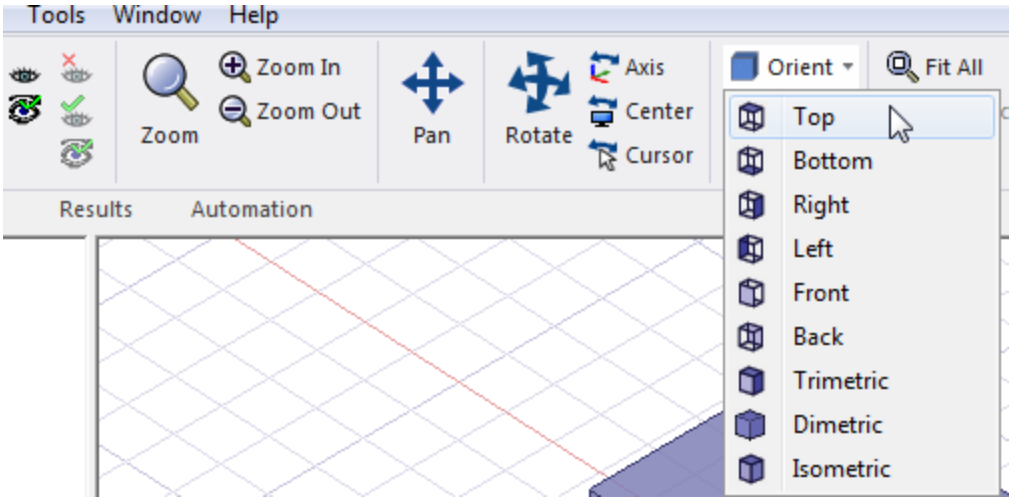
Modifying the View Orientation

There are several places you can access commands to change the orientation of the view in the *Modeler* window (that is, the viewing direction):

- Use the Orientation Gadget:



- Select the **Orient** drop-down Menu on the **View** ribbon tab and choose one of the predefined model orientations:



- Right-click in the Modeler window and, from the shortcut menu that appears, point to **View**> **Apply Orientation**, and select the desired viewing direction from the submenu. The choices include those shown in the preceding image plus any custom views you have defined (see next bullet).
- From the menu bar, click **View**> **Modify Attributes**> **Orientation List**.

A dialog box with orientation settings appears. It includes a table with the names of nine basic orientations (Top, Bottom, Right, Left, Front, Back, Trimetric, Dimetric, and Isometric), any additional orientations that you have added, and a section for managing orientations (adding/removing orientations or setting the default view). The table includes columns that show the viewing angles relative to the global axes and the equivalent vector components. These vector components are listed based on the global coordinate system and any applicable local coordinate systems.

1. Click **View>Modify Attributes>Orientation List**.

   The **Update View Orientation** dialog box appears. It includes a table of names of nine basic orientations (Top, Bottom, Right, Left, Front, Back, Trimetric, Dimetric, Isometric), any additional orientations that you have added, and a section for adding new orientations. The table includes columns that show the input angles, and the equivalent vector components.
2. Select a default orientation to the view, or create and apply a new orientation.
3. Click **Apply to View** for the selected view to appear in the view window.

   You can use the **Reset View Orientation** button to restore the view to the original angle.
4. Click **Make Default** if you want the selected viewing direction to be the initial viewing direction when a **Modeler** window is opened, either in the current project or future projects. Clicking the **Orient** button on the ribbon will apply default orientation.
5. Click **Close** to dismiss the dialog box.

   The orientation you set is saved with the design. New orientations assigned to other designs after this point do not affect this orientation.

Related Topics

Applying a Default View Orientation

Applying a New Orientation

Removing an Orientation

Using the Orientation Gadget

Applying a Default View Orientation

To apply a default viewing direction to the active view window:

1. Click **View>Modify Attributes>Orientation List**.

   The **Update View Orientation** dialog box appears.
2. Click one of the orientation names listed in the orientation list.
   - To view the associated vector components for the orientation you selected, select **Input vector components** under **Add Orientation to List**.

   The Vx, Vy, and Vz components are displayed in the boxes on the left.

   - To view the associated input angles for the orientation you selected, select **Input angles** under **Add Orientation to List**.

The psi, phi, and theta components of the selected orientation are listed in the boxes on the left.

3. Click **Apply to View**.

   The selected orientation is applied to the active view window.

   You can use the **Reset View Orientation** button to restore the view to the original angle.
4. Click **Close**.

Related Topics

Applying a New Orientation

Removing an Orientation

Applying a New View Orientation

To create and apply a new viewing direction to the active view window:

1. Click **View**>**Modify Attributes>Orientation**.

   The **Update View Orientation** dialog box appears.
2. To create a viewing direction that is based on a default viewing direction, click the existing orientation name in the viewing directions list.

   To create a viewing direction based on the current view in the **Modeler** window, click **Get Current View Direction**.
   - To modify the selected orientation's vector components, select **Input vector components** under **Add Orientation to List**, and then modify the values in the **Vx**, **Vy**, or **Vz** text boxes, and the Up vector boxes for Ux, Uy, and Uz.
   - To modify the selected orientation's input angles, select **Input angles** under **Add Orientation to List**, and then modify the values in the psi, theta, and phi text boxes.
3. Type a name for the new orientation in the **Name** box.
4. Click **Add/Edit**.

   The new orientation is added to the list of viewing directions.
5. Click **Make Default** if you want the new viewing direction to be the initial viewing direction when a **Modeler** window is opened in the current project or future projects.
6. Click **Close**.

Related Topics

Applying a Default View Orientation

Removing an Orientation

Removing an Orientation

To remove a viewing direction from the list in the orientation settings dialog box:

1. Click **View>Modify Attributes>Orientation**.

   The **Update View Orientation** dialog box appears.
2. Click the name of the viewing direction you want to delete from the orientation list.
3. Click **Remove**.
4. Click **Close**.

   The viewing direction is removed from the list.

Note	This operation cannot be undone.

Related Topics

Applying a Default View Orientation

Applying a New Orientation

Modifying the Lighting

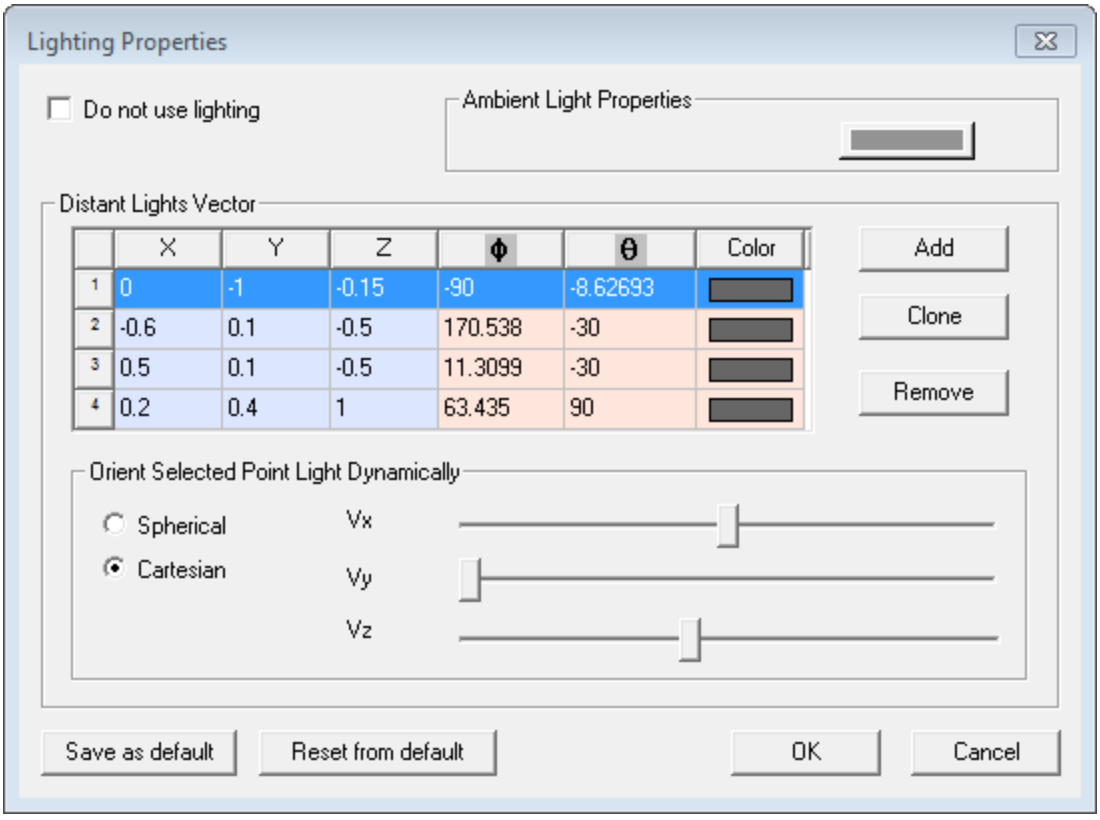
You have the option to emit the following types of light on a design:

- *Ambient* lighting surrounds the model evenly with light. All objects are lit evenly in every direction by a color of light that you specify.
- *Distant* lighting directs a ray of light at the model in a direction you specify. By default, two distant light vectors are in effect for every new view window.

To modify the lighting:

1. Click **View** > **Modify Attributes** > **Lighting** or, on the **View** tab of the ribbon, click on **Modify Attributes** and select **Lighting** from the drop-down menu.

   The **Lighting Properties** dialog box appears.



2. To turn off ambient and distant lighting, select **Do Not Use Lighting**.
3. To surround the model with light, click the **Ambient Light Properties** color button and then select a color for the surrounding light from the *Color* palette.
4. To modify the distant light on a model, do one of the following:
    - Add a new distant light by clicking **Add**.
    - Copy an existing distant light that you intend to modify by first selecting it in the **Distant Light Vectors** table and then clicking **Clone**.
    - Select a default distant light to modify by selecting it in the **Distant Light Vectors** table.
5. For the selected distant light vector, specify the vector direction:
    a. To modify the direction by specifying Cartesian coordinates, do one of the following:
        - Enter the new Cartesian coordinates in the **X**, **Y**, and **Z** boxes.
        - Use the **Vx**, **Vy**, and **Vz** sliders to specify the Cartesian coordinates dynamically.
    b. To modify the direction by specifying the spherical coordinates, do one of the following:
        - Enter the new spherical coordinates in the Φ (phi) and θ (theta) boxes.
        - Use the Φ and θ sliders to specify the spherical coordinates dynamically.

6. To revert to the default ambient and distant light settings, click **Reset**.
7. Click **Save As Default** if you want the new lighting settings to be the defaults for all **3D Modeler** windows, either in the current project or future projects.
8. Click **OK** to dismiss the dialog box.

   The lighting settings are saved with the design.

Note	New lighting applied to other designs after this point, including new default settings, do not affect these lighting settings.

Setting the Projection View

To modify the projection of model objects (the camera angle) in the view window:

1. Click **View>Modify Attributes>Projection**.

   The **Select Projection Type** window appears.
2. Select **Perspective** to change the camera angle of the view.
   - Move the slider to the right to increase the proximity, or widen, the view. Move the slider to the left to decrease the proximity, or flatten, the view.

   Objects that are closer appear larger relative to objects that are farther away.
3. Select **Orthographic** to view the model without distortion.

   The slider is disabled because a distortion scale is no longer applicable.
4. Click **Reset** to return the model to its original view.
5. Click **Close** to accept the projection setting and dismiss the window.

   The **Select Projection Type** window closes. The last view you specified in the projection window remains visible in the view window.

The projection view you set is saved with the design.

Note	New projection views assigned to other designs after this point do not affect this projection setting.

Setting the Background Color

To set the color of the background in the view window:

1. Click **View**>**Modify Attributes>Background color**.

   The **Select Background Color** window appears.
2. To assign a solid background color, do the following:
   a. Select **Plain Background**.
   b. Modify the background color in one of the following ways:
      - Click the **Background Color** button, and then select a color from the **Color** palette.
      - Click **OK**.
      - Use the RGB sliders under **Change View Color Dynamically** to specify the color's red, green, and blue values.

3. To assign a background color that gradually changes from one color to another, do the following:
   a. Select **Gradient Background**.
   b. Specify the background color at the top and bottom of the view window in one of the following ways:
      - Under **Select Background Type**, click the **Top Color** button, and select a color from the **Color** palette. Click **OK**. Then click the **BottomColor** button, and select a color from the **Color** palette. Then click **OK**.
      - Under **Change View Color Dynamically**, click **Top Color** or **Bottom Color**, and use the RGB sliders to specify the color's red, green, and blue values.
4. Click **Reset** to revert to the default background colors.
5. Click **Save As Default** if you want the new background color to be the background color for all **Modeler** windows in either the current project or future project.
6. Click **OK**.

The background color you set is saved with the design.

Note	New background color settings assigned to other designs after this point, including new default settings, do not affect this design.

Modifying the Coordinate System Axes View

What do you want to do?

- Show or hide the coordinate system axes.
- Show the coordinate system axes for selected objects.
- Enlarge or shrink the size of the coordinate system axes.
- Show or hide the triad axes.

Showing or Hiding the Axes

1. Click **View**>**Coordinate System**, then click one of the following:
   - **Hide** to hide the x-, y-, and z-axes in the active view window.
   - **Show** to display the x-, y-, and z-axes in the active view window.

Show the Axes for Selected Objects

1. Click **Tools**>**Options**>**Modeler Options**.
   This displays the Modeler Options dialog.
2. Select the **Display** tab.
3. Select **Show orientation of selected objects**.
4. Click **OK** to close the dialog.

Enlarging or Shrinking the Axes

1. Click **View>Coordinate System**, then click one of the following:
   - **Large** to display the x-, y-, and z-axes as extending to the edges of the active view window.
   - **Small** to display the x-, y-, and z-axes in a smaller size in relative to the edges of the active view window.

Showing or Hiding the Triad Axes

The triad is a secondary depiction of the coordinate system that appears at the lower right of the Modeler window. It shows the orientation of the currently selected working coordinate system. It can be shown or hidden separately from the selected coordinate system.

To show the triad:

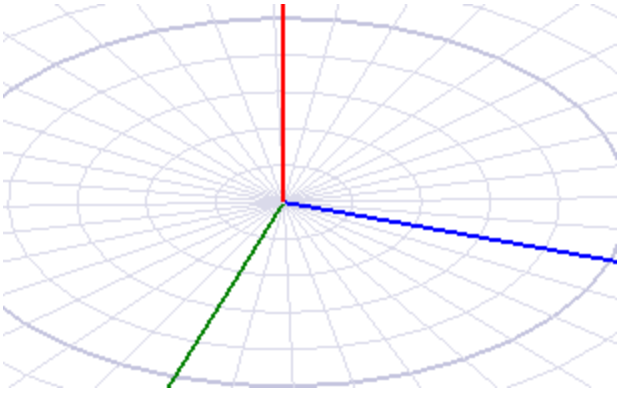
1. Click **View>Coordinate System>Triad**, then click one of the following:
   - **Hide** to hide the triad x-, y-, and z-axes at the lower right of the active view window.
   - **Show** to display the triad x-, y-, and z-axes in the lower right active view window.
   - **Auto** to generally hide the triad axes.

Choosing Grid Settings

The grid displayed in the **Modeler** window is a drawing aid that helps to visualize the location of objects.

For Cartesian grids, the location of points on the grid are defined by intersections of planes that are perpendicular to and along the x-, y-, and z-axes. The division (the distance between neighboring parallel planes perpendicular to the same axis) can be set.

For polar grids, the location of points on the grid are defined by intersections of planes that are perpendicular to the local radius and angle coordinates. The division (the distance between neighboring parallel planes perpendicular to the same radius and angle) can be set.



Grid spacing is set according to the current project's drawing units.

You can control the following aspects of the grid:

- Type (rectangular or circular)
- Style (dots or lines)
- Grid Extent
- Density
- Spacing
- Visibility
- Snap settings
- Grid plane

Setting the Grid Type

1. Click **View**>**Grid Settings**.

   The **Grid Spacing** window appears.
2. Select a **Grid type** for the active view window: **Cartesian** for a rectangular grid or **Polar** for a circular grid.

   The grid in the active view window is centered at the origin of the working coordinate system.

   - For Cartesian grids, define a coordinate by specifying its relative distance from the previously selected point in the **dX**, **dY**, and **dZ** text boxes.
   - For polar grids, define a coordinate by specifying its radius from the origin in the **R** text box and its angle from the x-axis in the **Theta** text box or its relative distance from the previously selected point in the **dR** and **dTheta** text boxes.

Setting the Grid Style

1. Click **View**>**Grid Settings**.

   The **Grid Spacing** window appears.

2. Select one of the following **Grid styles** for the active view window:

Dot	Displays each grid point as a dot.
Line	Displays lines between grid points.

3. To show a minimal grid around existing objects, check **Auto adjust grid extents**. To display a grid as a plane, uncheck **Auto adjust grid extents**.
4. Click **OK**.

Setting the Grid Density and Spacing

1. Click **View>Grid Settings**.

   The **Grid Spacing** window appears.
2. If you want to change the density of the grid in the active view window as you zoom in or out on objects, do the following:
   a. Select **Auto adjust density to**.
   b. Specify a distance between grid points by typing a value in the **pixels** box.

      The default is set to 30 pixels, which is generally the best setting for displaying objects.

   The distance between grid points will increase and decrease proportionately as you zoom in and out in the active view window.
3. If you do not want the grid density to change when you zoom in or out, but instead want to specify a constant grid spacing, do the following:
   a. Clear the **Auto adjust density to** option.
   b. Specify the grid's spacing in the active design's units.

      If you selected a Cartesian grid type, type the values of **dX**, **dY**, and **dZ**. These values represent the difference between one grid point and the next in the x, y, and z directions, respectively.

      If you selected a polar grid type, type the values for **dR** and **dTheta**. **dR** represents the difference between each radius. **dTheta** is the difference between angles.
4. Click **OK**.

Setting the Grid's Visibility

To hide the grid, click the Draw ribbon's Grid icon:  . Click it again to show the grid.

Alternatively:

1. Click **View**>**Grid Settings**.

   The **Grid Spacing** window appears.

2. To set Grid visibility:
    - Select **Show** to make the grid always visible in the active 3D Modeler window.
    - Select **Hide** to make the grid always invisible in the active 3D Modeler window.
    - Select **Auto** so the grid appears only while you are drawing an object in the 3D Modeler window.
3. Click **OK**.

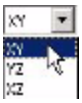
Related Topics

Setting the Grid Plane

Setting the Grid Plane

To specify the plane on which you want to display the grid in the active view window, do one of the following:

- Click **Modeler>Grid Plane**, and then select a grid plane: **XY**, **YZ**, or **XZ**.
- Select a grid plane from the pull-down list on the **Draw** ribbon.



Related Topics

Choosing Grid Settings

Selecting Items in the Modeler Window

To modify or learn more about an item's properties, you must first select it. All commands you choose while an item is selected are performed on or in reference to the selected item.

What selection mode do you want to use?

- Select Objects.
- Select Faces.
- Select Edges.
- Select Vertices.
- Select Multi (a mode for selecting objects, faces, edges or vertices)
- Coordinates in the drawing space.
- Select By Area
- Select by Variable
- Selecting Objects by Name
- Select by History Tree Folder
- Selecting the Face or Object Behind

Selecting Objects

By default, the modeler is in *object selection mode*. Click an object in the view window or an object name in the history tree to select it. All other objects become relatively transparent.

When the mouse hovers over an object in the view window, the object is highlighted. This indicates that it will be selected when you click it. Selected objects become the color specified under the **Display** tab of the **Modeler Options** dialog box.

Tooltips, as you hover the cursor over an entity, indicate the type/ID of entity (object name in the case of objects, Face_id in the case of faces, and so on). This feature helps you distinguish between face-of-sheet-object pick versus sheet-object pick.

If the modeler is not currently in object selection mode, you can switch to it using one of the following methods:

- Press the shortcut key **O**.
- Right-click in the view window, and then click **Select Objects**.
- Click **Edit>Selection Mode>Objects**.
- Select **Object** from the **Select:** pull-down list on the **Model** ribbon.

Related Topics

Selecting Several Objects
Selecting Objects by Name
Selecting All Faces of an Object
Selecting Objects in the History Tree
Creating an Object List
Selecting the Face, Edge, Vertex, or Object Behind
Select Edges.
Select Vertices.
Select Multi (a mode for selecting objects, faces, edges or vertices)
Select By Area
Selecting Objects and Surfaces that Lie Inside Other Objects
Clearing a Selection

Selecting Several Objects

1. If you are selecting objects in the Modeler window make sure that the modeler is in object selection mode by pressing the shortcut key **O**. You can always select objects in the History tree.
2. Select multiple objects in one of the following ways:
    - Hold down **Ctrl,** and click the objects in the view window that you want to select.
    - Hold down **Ctrl**, and click the object names in the history tree that you want to select.

- In the History tree, select a range of objects by first clicking one object to select it, and then Shift-click to extend the selection of visible items.
- In the History tree, under Lists, select AllObjects. This is an automatically created list that lets you selects all objects.
- Click **Edit>Select All** to select all objects that were drawn in the active view window, including objects that are not currently visible.
- Click **Edit>Select All Visible** (or press **Ctrl**+**A**) to select all objects that are visible in the active view window.
- Right click on a History tree icon for Model, Group, object, category such as Solid or Sheet, or material to perform context-sensitive selection. See Select Objects in the History Tree.

Selected objects become the color that is specified for selected objects under the **Display** tab of the **Modeler Options** dialog box. Use **Tools>Options>Modeler Options** to display the dialog and set the default color. By default, the selected objects are opaque and all other objects become relatively transparent.The settings for the relative opacity and transparency of selected and non-selected objects also appear in the **3D UI Options** dialog box. Use **View>Options** to display the **3D UI Options** dialog.

To deselect all objects, do one of the following:

- Click **Edit>Deselect All**.
- Press **Ctrl+Shift+A**.

Related Topics

Selecting Objects by Name
Selecting All Faces of an Object
Creating an Object List
Selecting the Face or Object Behind
Select Edges.
Select Vertices.
Select Multi (a mode for selecting objects, faces, edges or vertices)
Select By Area
Selecting All Objects in a History Tree Folder

Selecting Objects by Name

1. Make sure that Maxwell is in object selection mode by pressing the shortcut key **O**.

2. Click **Edit>Select Objects>By Name,**

   or on the **Model** ribbon, select **Object** from the **Select:** drop-down menu, then click **Select by Name**.

   The **Select Object** dialog box appears.
3. In the **Name** list, click the name of the object you want to select. Use the **Ctrl** key to select more than one.
   - Alternatively, type the name of an object you want to select in the empty text box, click the right-arrow button, and select either **Select** or **Deselect**.
4. Click **OK**.

   The object is selected.

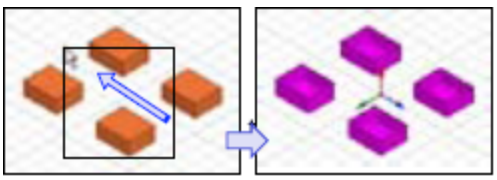
Related Topics

Selecting Several Objects
Selecting Objects by Name
Selecting All Faces of an Object
Creating an Object List
Selecting the Face or Object Behind
Select Edges.
Select Vertices.
Select Multi (a mode for selecting objects, faces, edges or vertices)
Select By Area

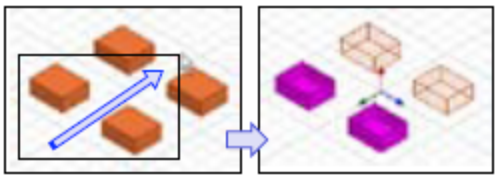
Select By Area

You can select items by area in the Modeler window by clicking and rubber-band dragging around objects. Rubber-band selection works differently depending on the drag direction.

- From right to left: Selects all the items that are wholly or partly enclosed within the rubber band.



- From left to right: Selects all the items wholly enclosed within the rubber band.

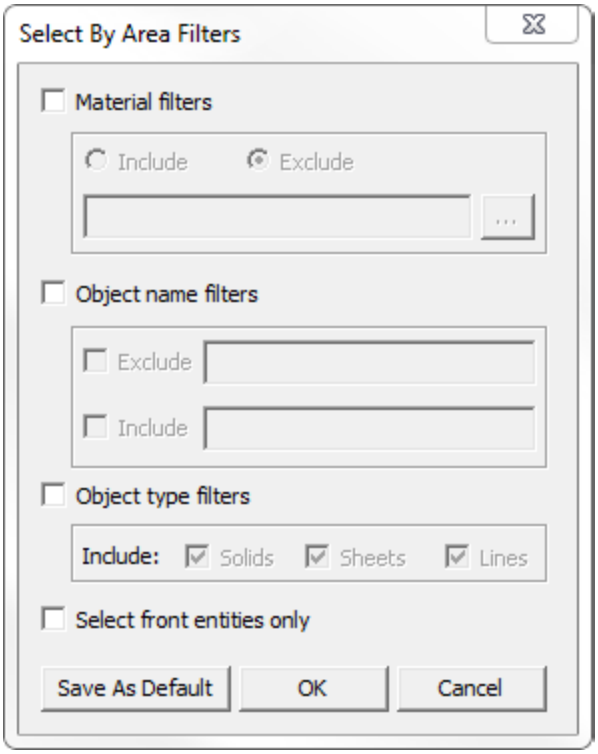


Select By Area works with Selection mode for Objects, Faces, Edges, and Vertices, but not for Select Multi. By default, only items with external surfaces are selected.

However you can control which objects to include or exclude from area selection based on material, object names, or object types.

To do this:

1. Click **Edit>Select By Area Filter** to display this dialog:



2. Check **Material filters** to enable the **Include** and **Exclude** radio buttons. Use the text field to specify filters by name, or use the ellipsis [...] button to display the Materials manager for selections.
3. Check **Object name filters** to enable the **Exclude** and **Include** check boxes, and text fields in which you can specify object names.
4. Check **Object type filters** to enable the check boxes for including **Solids**, **Sheets**, and/or **Lines**.

5. Check **Select front entities only** to select only objects that are in front.
6. If you click **Save As Default**, the settings persist for the project until you change the settings and **Save as Default** again.
7. Click **OK** to close the dialog.

Now, when you left-click and drag around an area, those objects which meet the filter criteria are highlighted in the Modeler window, and those objects are shown as selected in the History tree.

Related Topics

Selecting Items in the Modeler Window

Selecting Multiple Objects

Selecting Objects by Name

Selecting All Faces of an Object

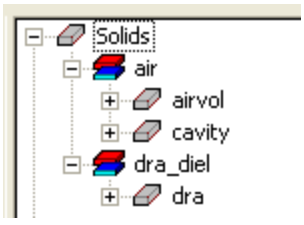
Creating an Object List

Selecting All Objects in a History Tree Folder

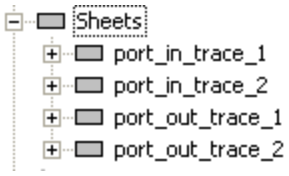
The history tree organizes objects in several categories. Right click on the group for solids, sheets, lines, non-model objects, or unclassified objects to see a shortcut menu command that lets you **Select All** members of that category.

Under each folder, you can also **Select All** for objects of the same the material or parts of the same assignment at once. When you right-click on a solid folder, the menu lets you check whether to organize objects by assignment. For example, this history tree has Solids organized by object.

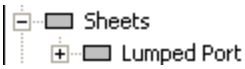
This history tree has Organize Objects by Material selected.



This history tree organizes Sheet objects by object.



This history tree has Organize Sheets by Assignment selected.



Selected objects are highlighted in the modeler window view area.

Related Topics

Selecting Multiple Objects
Selecting Objects by Name
Select By Area
Selecting All Faces of an Object
Creating an Object List
Setting the Default Color and Transparency of Selected Objects
Setting the Default Color of Highlighted Objects
Working with the History Tree

Setting the Default Color and Transparency of Selected Objects

To set the color of objects when they are selected:

1. Click **Tools>Options>Modeler Options**.
   The **Modeler Options** dialog box appears.
2. Click the **Display** tab.
3. Click **Select** on the **Default color** pull-down list.
4. Click the color button beside the **Default color** pull-down list.
   The **Color** palette appears.
5. Select a color from the **Color** palette, and then click **OK**.
   Any objects you select after this point will temporarily become the default color you selected.

By default, Maxwell shows selected objects as nearly opaque and shows non-selected objects as nearly transparent. This features helps you distinguish between selected and non-selected objects.

To set the transparency of selected and non-selected objects:

1. Click **View>Options**.

   The **3D UI Options** dialog appears. The **When there is a selection** region contains check boxes for setting the transparency for selected and non-selected objects.
2. Click the check box for the value you want to change.

   This enables the value field. The default transparency for selected objects is 0.1, which makes them almost opaque. The default transparency for non-selected objects is 0.9, which makes them highly transparent.
3. Enter a new value, and click **OK** to apply the new transparency values.

Setting the Default Color of Highlighted Objects

1. Click **Tools**>**Options>Modeler Options**.

   The **Modeler Options** dialog box appears.
2. Click the **Display** tab.
3. Click **Highlight** on the **Default color** pull-down list.
4. Click the color button beside the **Default color** pull-down list.

   The **Color** palette appears.
5. Select a color from the **Color** palette, and then click **OKto return to the Modeler Options** dialog box.
6. Click **OK**.

   Outlines of all object you hover over temporarily become the default color you specified above.

Creating an Object List

Create an object list when you want to define a list of objects. Creating an object list is a convenient way to identify and select a group of objects for a field plot or calculation. Objects in a list can still be treated as separate objects. The same object can be included in several different lists.

To create an object list:

1. If you are selecting in the Modeler window, make sure that the modeler is in object selection mode by pressing the shortcut key **O**. This is not necessary for selecting in the History tree.
2. Select the objects you want to include in the list.
3. Click **Modeler>List>Create>Object List**.

   The object list is created with the default name Objectlist*n*. It is listed in the history tree under **Lists**. Selecting an object list displays the properties of that list in the Properties window. One of the properties is a list of objects contained in the list.

   To rename the Object list, edit the Name property in the **Properties** window for the list. Object lists are sorted in alphanumeric order.

The object list is treated as one volume when you are plotting and performing fields calculations. It is listed in the **Geometry** window of the Fields Calculator when you select **Volume**.

There is an automatically created list called AllObjects. Selecting it selects all objects. If a list contains mixed types of geometry, for example, volume and sheet objects, the volume calculation only uses the geometry of the highest dimension in plots or integral, and so forth.

Example: To plot the E-field on a surface formed by the intersection of the xy-plane and several objects, first define a list of these objects. Then, when plotting fields, select the object list name from the **Geometry** window of the Fields Calculator. Fields are plotted only at the intersection of the plane and the objects in the list.

Related Topics

Reassigning Objects to Another Object List

Using or Viewing List Objects

Reassigning Objects to Another Object List

You can assign objects after you have created object lists. Creating an object list is a convenient way to identify and select a group of objects for a field plot or calculation. Objects in a list can still be treated as separate objects. The same object can be included in several different lists.

To reassign objects to an existing object list:

1. If you are selecting in the Modeler window, make sure that the modeler is in object selection mode by pressing the shortcut key **O**. This is not necessary for selecting in the History tree.
2. Select the objects you want to reassign.
3. Click **Modeler>List>Reassign**

   A dialog with the existing object lists is displayed. (They appear in the history tree under **Lists**.) One of the Properties in for the List shows the objects contained in the list.
4. Select the list to which you want to assign the select object and click **OK**.

   The object is reassigned to the selected list, replacing previous list members. The Objects Property in for the List shows the objects contained in the list.

The object list will be treated as one volume when you are plotting and performing fields calculations. It will be listed in the **Geometry** window of the Fields Calculator, when you select **Volume**.

Related Topics

Creating an Object List

Using or Viewing List Objects

Using or Viewing List Objects

To view the objects included in an Object list:

1. In the model tree, expand the **Lists** tree.
2. Right-click the list you want to select, and click **Select Assignment**.

   The objects that are included in that list are highlighted in the modeler, and the properties appear in the desktop.

To view the properties of the object list (including a list of the objects included):

1. In the model history tree, expand the **Lists** tree.
2. Under **Lists**, right-click the list object you want to view, and click **Properties**.
   The **Properties** window appears for that object list. The objects included are listed in the **Objects** row.
3. Click **OK** or **Cancel** to close the **Properties** window.

To use an object from a list in another operation:

1. In the model tree, expand the **Lists** tree.
2. Right-click the list you want to select, and click **Select Assignment**.

   The objects that are included in that list are highlighted in the modeler, and the properties appear in the desktop.
3. Select any other objects you want to use in the operation.
4. Complete the operation.

For example, you could select an object list and another object, and then specify one of the boolean commands (such as unite or subtract).

Selecting Faces

If the modeler is in face selection mode, click an object face in the view window to select it. To select multiple faces, hold the **Ctrl** key as you click the faces. You also have the option to create face lists, which define a list of object faces, or you can make face selections from a Face ID list in the By Face dialog.

Switch to face selection mode using one of the following methods:

- Click **Edit>Selection Mode>Faces**.
- Select **Face** from the **Select** pull-down list on the **Model** ribbon.
- Right-click in the view window, and then click **Select Faces**. (Not available when working on a 2D Design)
- Press the shortcut key **F**. (Not available when working on a 2D Design)

You can also select faces in the Select Multi mode.

When the mouse hovers over a face in the view window, that face is highlighted, which indicates that it will be selected when you click. Selected faces become the color specified under the **Display** tab of the **Modeler Options** dialog box. All other objects and faces become relatively transparent.

To select faces touching the current selection:

- Click **Edit>Extend Selection>Select Connected Faces**.

To select faces that touch each other:

- Click **Edit>Extend Selection>Select Face Chain**. This option allows faces that are part of a "protrusion" to be selected.

You can also use the **By Face** dialog to select from a list of faces associated with an object:

1. To use the dialog box, no objects should be selected to start.
2. Click **Edit**>**Selection Mode**>**Faces** or on the Model ribbon, Select Face from the drop-down menu; then click the Select by Name  icon.

   This displays the **Select Face** dialog. This contains a list of the available objects.
3. Select an object in the Object Name list.

   The Face ID list is then populated with the faces in that object.
4. Selecting a face ID from the list highlights the face in the 3D window. Use Ctrl-click to select additional faces, or shift-click to select a range of faces.

Related Topics

Selecting All Faces of an Object
Selecting the Face, Edge, Vertex, or Object Behind
Selecting Faces by Name
Selecting Faces by Plane
Creating a Face List
Face Selection
Select Edges.
Select Vertices.
Select Multi (a mode for selecting objects, faces, edges or vertices)

Selecting All Faces of an Object

This option allows you to select all faces of an object after first selecting the object or a single face.

Note	This options is not available when working on a 2D Design.

1. Optionally, select the object (or objects, faces, edges or vertices) with the faces you want to select.
2. Switch to face selection mode by pressing the shortcut key **F**.
3. If an object is not selected, click a face on the object of interest.
4. Click **Edit>Extend Selection>All Object Faces**.
   - Alternatively, right-click in the view window, and then click **Extend Selection>All Object Faces** on the shortcut menu.
   - As another alternative, you can use the Select pull-down menu on the Model ribbon.

   All the faces of the object are selected. If you selected multiple objects, all faces of those objects are selected.

Related Topics

Selecting Faces
Selecting the Face or Object Behind
Creating a Face List

Selecting Faces by Name

1. Make sure that the modeler is in face selection mode by pressing the shortcut key **F**.
2. Click **Edit>Select Objects>By Name**,

   or on the **Model** ribbon, select **Face** from the **Select:** drop-down menu, then click **Select by Name**

   The **Select Face** dialog box appears.
3. In the **Object name** list, click the name of the object with the face you want to select.

   The object's faces are listed in the **Face ID** column.
4. Click the face you want to select in the **Face ID** column. You can select more than one.

   The face is selected in the view window.
5. Click **OK**.

Related Topics

Selecting Faces
Creating a Face List

Selecting Faces by Plane

To select a face that is aligned with a global plane, use one of the following two methods.

1. Make sure that the modeler is in face selection mode by pressing the shortcut key **F**.
2. In the History Tree, expand the **Planes** icon. Left-click on a plane (Global:XY, Global:YZ, or Global:XZ) to display the selected global plane.
3. On the **Edit** menu, point to **Extend Selection**, and then click **Faces on Plane**.

   The selected faces are highlighted.

Alternative method:

1. In the History Tree, expand the **Planes** icon.
2. Right-click on a plane (Global:XY, Global:YZ, or Global:XZ) to select the global plane and display a pull-down menu.
3. On the pull-down menu, click **Faces on Plane**.

The selected faces are highlighted.

Related Topics

Selecting Faces

Creating a Face List

Select Multi (a mode for selecting objects, faces, edges or vertices)

Face Selection

While working on analyzing complex objects, it is sometimes useful to examine faces, edges and vertices. In particular it is useful to find the connected faces for a face or edge or vertex, connected edges for a face/edge/vertex and connected vertices for a face/edge/vertex. The additional selection modes are available under **Edit->Selection Mode**, and **Edit>Extend Selections**.

- **Select Face Chain** selects faces that touch each other. It allows faces that are part of a "protrusion" to be selected.
- **Select Connected Faces** selects faces connected to the current selection.
- **Select Connected Edges** selects the edges of the selected face or faces.
- **Select vertices** selects the vertices of the selected face or faces.

Related Topics

Selecting All Faces of an Object

Selecting the Face or Object Behind

Selecting Faces by Name

Selecting Faces by Plane

Creating a Face List

Select Edges.

Select Vertices.

Select Multi (a mode for selecting objects, faces, edges or vertices)

Creating a Face List

Create a face list when you want to define a list of object faces. Creating a face list is a convenient way to identify and select a specific set of surfaces for a field plot or calculation. The same face can be included in several different lists.

To create a face list:

1. Make sure that the modeler is in face selection mode by pressing the shortcut key **F**.
2. Select the object faces you want to include in the face list.
3. Click **Modeler>List>Create>Face List**.

The face list is created. It is listed in the history tree under **Lists**. The default name is Facelist*n.* The lists appear in alphanumeric order. To change the name of a face list (for example, to a name describing the listed faces as ports or boundaries), select the list in the History Tree and **Edit Properties**. Editing the Name property changes the name. If necessary, the list order in the History tree changes for the new name.

The face list is treated as one selection of surfaces when you are plotting and performing fields calculations. The face list will be listed in the **Geometry** window of the Fields Calculator when you select **Surface**.

Selecting Edges

If the modeler is in edge selection mode, simply click an object's edge in the view window to select it. To select multiple edges, hold the **CTRL** key as you click the edges.

When the mouse hovers over an edge in the view window, that edge is highlighted, indicating it is click-able. Selected edges become the color specified under the **Display** tab of the **Modeler Options** dialog box. All other objects become relatively transparent.

Switch to edge selection mode using one of the following methods:

- Click **Edit>Selection Mode>Edges**.
- Select **Edge** from the pull-down list on the **Model** ribbon.
- Right-click in the modeler window, and then click **Selection Mode>Edges**.
- Press the shortcut key **E**.

Edit>Extend Selection offers additional edge selections such as:

- **Select Edge Chain** selects the edges that touch the selected edge.
- **Select Connected Faces** selects faces touching to the current selection.
- **Select Connected Edges** selects the edges that touch the current selection.
- **Select Connected Vertices** selects the vertices of the selected edge or edges.

Related Topics

Moving Edges Along the Normal

Selecting All Faces of an Object

Selecting the Face or Object Behind

Selecting Faces by Name

Selecting Faces by Plane

Creating a Face List

Select Edges

Selecting All Edges of an Object

Select Vertices

Select Multi (a mode for selecting objects, faces, edges or vertices)

Selecting All Edges of an Object or Face

This option allows you to select all edges of an object or face after first selecting the object or face.

1. Select the object or face with the edges you want to select.

   You may also select a single edge of the object or face.
2. Click **Edit>Extend Selection>All Object Edges** or **Edit>Extend Selection>All Face Edges**.
   - Alternatively, right-click in the view window, and then click **Extend Selection>All Object Edges** or **Extend Selection>All Face Edges** on the shortcut menu.

   All the edges of the object or face are selected. If you selected multiple objects, all edges of those objects are selected.

Related Topics

Selecting Faces

Selecting the Face or Object Behind

Creating a Face List

Face Selection

Selecting Vertices

If the modeler is in vertex selection mode, simply click an object's vertex in the view window to select it. To select multiple vertices, hold the **Ctrl** key as you click the vertices.

When the mouse hovers over a vertex in the view window, that vertex is highlighted, which indicates that it will be selected when you click. Selected vertices become the color specified under the **Display** tab of the **Modeler Options** dialog box. All other objects become relatively transparent.

Switch to vertex selection mode using one of the following methods:

- Click **Edit>Selection Mode>Vertices**.
- Select **Vertex** from the **Select:** pull-down list on the **Model** ribbon.
- Press the shortcut key **V**.

Selecting a vertex enables the following selections on the **Edit>Extend Selection** menu and the model view window **Extend Selection** context menu:

- **Select Connected Faces** selects faces touching to the current selection.
- **Select Connected Edges** selects the edges that touch the current selection.
- **Select Connected Vertices** selects the vertices of edges that touch the current selection.

Related Topics

Selecting All Faces of an Object

Selecting the Face or Object Behind

Selecting Faces by Name
Selecting Faces by Plane
Creating a Face List
Select Edges
Select Vertices
Select Multi (a mode for selecting objects, faces, edges or vertices)

Selecting Multi (a Mode for Selecting Objects, Faces, Vertices or Edges)

The Select Multi mode permits you to select objects, faces, vertices, or edges, depending on where you click. This very useful in conjunction with Measure Mode, for measuring the distances between different entities. Enter Select Multi mode by one of the following methods:

- Press the shortcut key **M**.
- Right-click in the view window, and then click **Selection Mode>Select Multi**.
- On the **Edit** menu, point to **Selection Mode**, and then click **Multi**.
- Select **Multi** from the **Select:** pull-down list on the **Model** ribbon.

With Multi mode active:

- To select a vertex, click near a vertex, within 10 pixel radius.
- To select an edge, click near an edge (and 10 pixels away from vertex).
- To select an object, click little farther from edge, between 10 and 20 pixels.
- To select a face, click anywhere else on the interior of face.

Tooltips, as you hover the cursor over an entity, indicate the type/ID of entity (object name in the case of objects, Face_id in the case of faces, and so on). This feature helps you distinguish between face-of-sheet-object pick versus sheet-object pick.

By holding down the Ctrl key, you can make multiple selections.

Related Topics

Controlling the Selection in Multi Mode

Controlling the Selection in Multi Mode

You can control the behavior of this mode by clicking **Edit>Selection Mode>Select Multi Mode Settings**. This displays a dialog with check boxes for Object, Face, Edge, and Vertex. Unchecking a box cancels the selection behavior for that category.

You can also control the behavior of this mode on the **Model** ribbon tab by selecting the desired mode from the **Select:** drop-down list.

Related Topics

Selecting All Faces of an Object

Selecting Groups and Submodels

With group selection mode, the parent group of selected object is selected. Selecting group highlights all children under the group in the Modeler window. If you click an object that has not been assigned to a group, nothing is selected.To select multiple groups, hold the **CTRL** key as you click the groups.

To set the selection mode to Group:

- From the menu bar, click **Edit**>**Selection Mode**>**Groups**
- Press the "**G**" key to enter group selection mode.
- From the 3D Modeler window context menu, select **Selection Mode>Group**

If your design contains submodels, you can set the selection mode to Submodel.

- From the menu bar, click **Edit**>**Selection Mode**>**Submodel**
- Press the "**U**" key to enter Group selection mode.
- From the 3D Modeler window context menu, select **Selection Mode>Submodel**

Related Topics

Selecting the Face or Object Behind

Select Multi (a mode for selecting objects, faces, edges or vertices)

Clearing a Selection

To clear an object, face, edge, or vertex selection, do one of the following:

- Click the view window at a location where no objects exist.
- To clear an object selection, click a point away from the object name in the history tree.
- Click **Edit>Deselect All**.
- Press **Ctrl+Shift+A**.

  The items are no longer selected.

Selecting the Face, Edge, Vertex, or Object Behind

To select the face, edge, vertex, or object behind another selected face, edge, vertex, or object, do one of the following:

- Click **Edit>Next Behind**.
- Right-click in the view window and click **Next Behind**.

  When there are multiple faces behind, the one selected is relatively close to where you right-click.
- Press the shortcut key **B**.

  When there are multiple faces behind, the one selected is relatively close to the cursor.
- Press **Ctrl+B**.

  This option is useful when you are trying to select a face, edge, vertex, or object that is in the interior of a model, or when you do not want to change the model view to select an item.

Related Topics

Selecting Objects and Surfaces that Lie Inside Other Objects

Selecting the Edges of the Problem Region

If you have excluded the background object from your model, do not assign boundary conditions or sources to it.

Selecting Objects and Surfaces That Lie Inside Other Objects

To select objects and surfaces that lie inside other objects (such as an object that lies within an air box, a conductive shield, or the background object), do one of the following:

- Make the objects on the outside of the model invisible using the **View>Visibility** commands. This is useful when you want to select objects using the mouse. Since the mouse cannot select invisible objects, you can select the interior surfaces or objects by clicking on them.
- Use the **Edit>Select Objects>By Name** command to select objects or surfaces inside the model.
- Use the **Next Behind** command on the shortcut menu. This selects the object that lies behind the one you initially selected. This command does nothing if no objects have previously been selected.

Related Topics

Selecting the Face or Object Behind

Using the Mouse to Select Objects

To select objects or surfaces, do one of the following:

- Click the object directly.
- Right-click an object or surface, and use one of the **Select** commands on the shortcut menu.
  The following commands appear on the shortcut menu:
  - **Select Objects**
  - **Select Faces**
  - **Next Behind**: Use this command to select the object or face that lies behind the currently selected object or face. This command chooses objects or faces depending on the graphical pick mode. **Next Behind** does nothing if no object has previously been selected or if the object you select has nothing behind it. You can also use the keyboard shortcut **B**.
  - **All Object Faces**
  - **Faces on Plane**

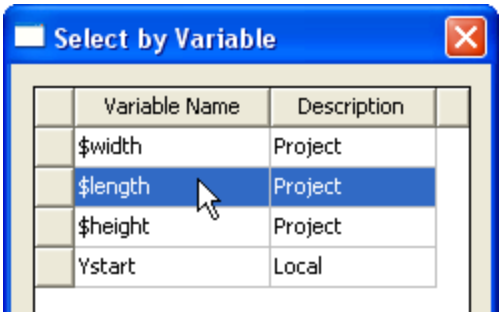
The snap mode defines how items are selected by the mouse. By default, **Grid** and **Vertex** snaps are enabled.

Selecting Objects by Variable

You can select an object based on a variable that affects it. If your design includes variables, you do so as follows:

1. Click **Edit>Select Objects>By Variable**.

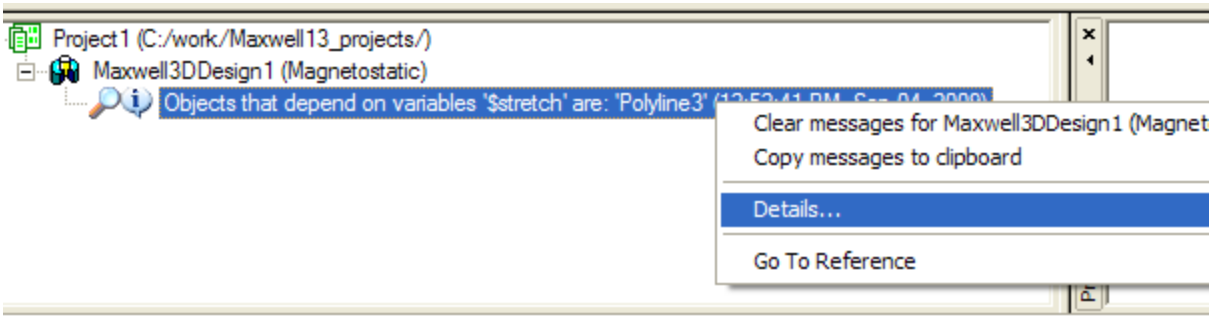
   This displays the Select by Variable dialog, which lists the variables in your design. Both columns are sortable by clicking the header. You can resize and move the dialog box. When you next open it, it uses that size and location.



2. Select the variable of interest, and click **OK**.

   The dialog closes, and the object affected by the variable is highlighted in the Modeler window.

   The Message window contains a reference that you can select and use to go to the affected object.



If you execute the command again, without clearing the current selection(s), the additional object can be highlighted. You can resize and move the dialog box. When you next open it, it uses that size and location.

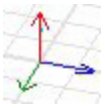
Assigning Coordinates to New Objects

When you insert and draw new 3D, 2D, or 1D objects in the geometry model, you need to define coordinate systems and locations of points, distances between objects, and other geometry items.

- Setting the Reference Point
- Defining Cartesian Coordinates
- Defining Cylindrical Coordinates
- Defining Spherical Coordinates
- Defining Relative Coordinates
- Defining Absolute Coordinates

Setting the Reference Point

When you draw objects, the cursor's location is always relative to a reference point. The reference point is displayed with a mini xyz-axis:



To change the reference point:

1. Select the drawing command to use.

   The Measure Data dialog opens. As you move the cursor over the modeler window, the top line in the measure dialog shows the coordinates of the current reference point.
2. Move the cursor to the desired reference point and press **Ctrl+Click** or right click and select **Set Reference Point** from the short cut menu.

   This moves the reference point marker to the new location. The **Measure Data** dialog updates. The coordinates boxes in the Status bar change to accept relative distance

information. If you choose, rather than setting the reference point with the cursor, you can press Tab to activate a text cursor in the status bar fields, and enter coordinates directly.

Related Topics

Assigning Coordinates to New Objects
Choosing the Movement Mode
Choosing Snap Settings
Drawing Objects

Defining Cartesian Coordinates

When drawing an object, define a point using Cartesian coordinates by typing its distance from the origin in the **X**, **Y**, and **Z** boxes, respectively. When defining a second point, specify its distance from the previously defined point in the x, y, and z directions in the **dX**, **dY**, and **dZ** boxes, respectively.

1. Select the desired drawing command.
2. Select **Cartesian** from the pull-down list on the status bar.
3. Type the point's x-, y-, and z-coordinates in the **X**, **Y**, and **Z** text boxes.

Hint	Press **Tab** to move from one coordinate text box to the next. Press **Shift+Tab** to move to the previous coordinate text box.

   - Alternatively, click the point in the view window.
4. When drawing objects other than polylines, the second point you select is relative to the first point. Type the second point's distance from the previously selected point in the **dX**, **dY**, and **dZ** boxes.

Related Topics

Selecting Cylindrical Coordinates
Selecting Spherical Coordinates

Defining Cylindrical Coordinates

To define a point using cylindrical coordinates, specify the point's radius, measured from the origin, in the **R** box, the angle from the x-axis in the **Theta** box, and the distance from the origin in the z direction in the **Z** box. When defining a second point, specify its distance from the previously defined point in the in **dR**, **dTheta**, and **dZ** boxes.

1. After clicking the desired drawing command, select **Cylindrical** from the pull-down list on the status bar.
2. Type the point's r-, theta-, and z-coordinates in the **R**, **Theta**, and **Z** boxes.

Hint	Press **Tab** to move from one coordinate box to the next. Press **Shift+Tab** to move

	to the previous coordinate box.

- Alternatively, click the point in the view window.

3. When drawing objects other than polylines and helices, the second point you select is relative to the first point. Type the second point's distance from the previously selected point in the **dR**, **dTheta**, and **dZ** boxes.

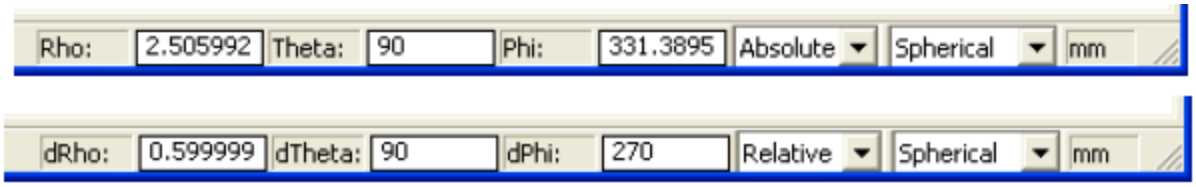
Related Topics

Selecting Cartesian Coordinates

Selecting Spherical Coordinates

Defining Spherical Coordinates

To define a point in spherical coordinates, specify the point's radius, measured from the origin, in the **R** box, the angle from the z-axis in the **Theta** box, and the angle rotated counter-clockwise (CCW) from the x-axis in the **Phi** box. When defining a second point, specify its distance from the previously-defined point in the in **dR**, **dTheta**, and **dPhi** boxes.



Note	Even though you are inputting spherical coordinates, all data is internally stored in Cartesian coordinates.

To define a point in spherical coordinates:

1. After clicking the desired drawing command, select **Spherical** from the pull-down list on the status bar.
2. Type the point's r-, theta-, and phi-coordinates in the **R**, **Theta**, and **Phi** boxes.

Hint	Press **Tab** to move from one coordinate box to the next. Press **Shift+Tab** to move to the previous coordinate box.

- Alternatively, click the point in the view window.

3. When drawing objects other than polylines and helices, the second point you select is relative to the first point. Type the second point's distance from the previously selected point in the **dR**, **dTheta**, and **dPhi** boxes.

Related Topics

Selecting Cartesian Coordinates

Selecting Cylindrical Coordinates

Using Absolute Coordinates

When entering a point's coordinates, you can specify them in *absolute* or *relative* coordinates. Absolute coordinates are relative to the working coordinate system's origin (0, 0, 0). This is the default setting for the first point you select after clicking a drawing command. Relative coordinates are relative to the reference point, or the previously selected point, represented by a small coordinate system on the user interface.

To enter a point's absolute coordinates:

1. Click the desired drawing command.
2. Select **Absolute** from the **Absolute/Relative** pull-down list on the status bar.
3. Specify the point's coordinates in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the appropriate boxes on the status bar.

Note	When drawing objects other than polylines and helices, by default, the second point you select is relative to the first point; **Relative** is automatically selected in the **Absolute/Relative** pull-down list in the status bar. Be sure to select **Absolute** from the **Absolute/Relative** pull-down list in the status bar if you want the second point to be relative to the working coordinate system.

Related Topics

Selecting Relative Coordinates

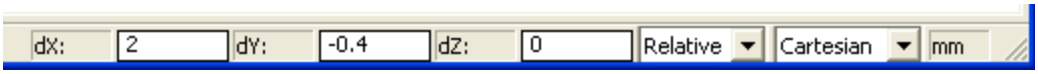
Using Relative Coordinates

When entering a point's coordinates, you can specify them in *absolute* or *relative* coordinates. Relative coordinates are relative to the reference point, or the previously selected point. Absolute coordinates are relative to the working coordinate system's origin (0, 0, 0).

To enter a point's relative coordinates:

1. Click the desired drawing command.
2. Select **Relative** from the **Absolute/Relative** pull-down list on the status bar.

   When you are in relative mode, the text boxes for a coordinate show an “d” before the coordinate description, to indicate “distance from” the working reference. For example:



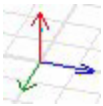
3. Specify the point's coordinates in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the appropriate boxes in the status bar.

Related Topics

Selecting Absolute Coordinates

Choosing the Movement Mode

When drawing objects, the cursor's location is always relative to a reference point. The reference point is displayed with a mini xyz-axis:



To change the reference point, move the cursor to the desired point and press **Ctrl+Enter**.

You can move the cursor to one of the following points:

- In the same plane as the reference point (in-plane movement mode).
- Perpendicular to the reference point (out-of-plane movement mode).
- If an object is present to snap to a point in 3D space (3D movement mode).
- Along the x-axis
- Along the y-axis
- Along the z-axis

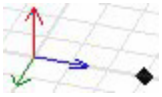
Changes you make to the movement mode persist until you change them again.

Moving the Cursor In Plane

To move the cursor to a point *on the same plane* as the reference point:

1. Click the desired drawing command.
2. Do one of the following:
   - Click **Modeler** > **Movement Mode**>**In Plane**.
   - In the model view window, right-click and select **Movement Mode**>**In Plane**.

   The next point you select will be on the same plane as the reference point.



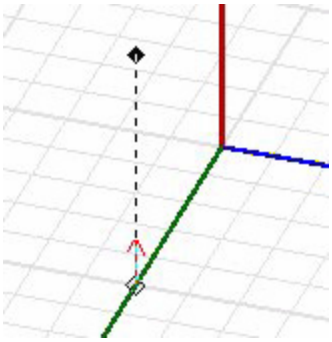
The cursor's location, displayed with a black diamond that indicates it has snapped to the grid, is on the same plane as the reference point.

Moving the Cursor Out of Plane

To move the cursor to a point on a line that is *perpendicular* to the drawing plane and that intersects the reference point:

1. After clicking the desired drawing command.
2. Do one of the following:
   - click **Modeler>Movement Mode>Out of Plane**.
   - In the model view window, right-click and select **Movement Mode**>**Out of Plane**.

   A dashed line is displayed between the reference point and the cursor's location, which is now on a line that is perpendicular to the drawing plane and that intersects the reference point.

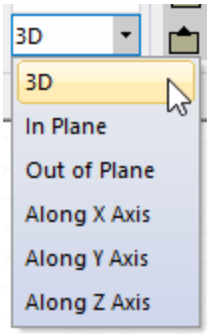


The cursor's location, displayed with a black diamond that indicates it has snapped to a grid, is on a line that is perpendicular to the drawing plane and that intersects the reference point.

Moving the Cursor in 3D Space

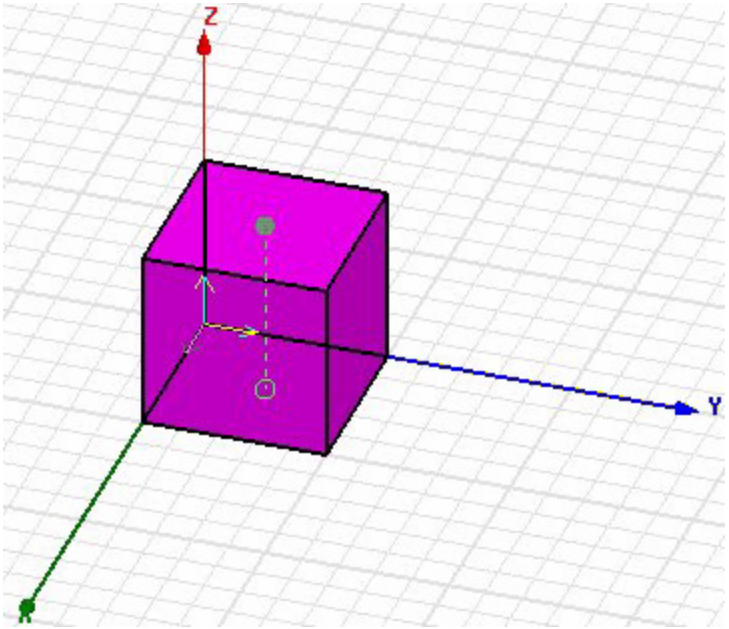
To move the cursor to a point in 3D space relative to the reference point:

1. Click the desired drawing command.
2. Do one of the following:
   - Click **Modeler>Movement Mode>3D**.
   - Click **3D** in the movement mode pull-down list on the **Draw** ribbon.



If one of an object's snapping centers is within snapping range, the cursor snaps to the nearest point in 3D space occupied by the object.

If any of the objects has a snapping center not within snapping range, the 3D movement mode is identical to the in-plane movement mode.



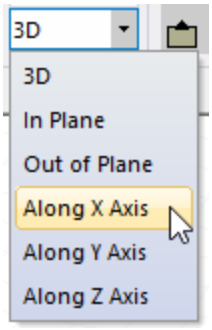
The cursor's location, displayed by a circle that indicates it has snapped to a face center, is (0.5, 0.5, 1.0), a point in 3D space.

Moving the Cursor Along the X-Axis

To move the cursor to a point away from the reference point in the x direction:

1. Click the desired drawing command.
2. Do one of the following:
   - Click **Modeler>Movement Mode>Along X Axis**.
   - Press and hold down the shortcut key **X**.

- Select **Along X Axis** from the **Movement mode** pull-down list on the **Draw** ribbon.

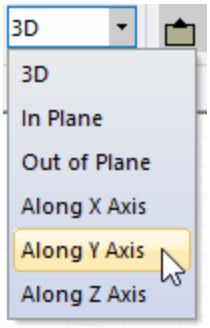


The next point you select will be on the same plane as the reference point in the positive or negative x direction.

Moving the Cursor Along the Y-Axis

To move the cursor to a point away from the reference point in the y direction:

1. Click the desired drawing command.
2. Do one of the following:
   - Click **Modeler>Movement Mode>Along Y Axis**.
   - Press and hold down the shortcut key **Y**.
   - Select **Along Y Axis** from the **Movement mode** pull-down list on the **Draw**ribbon.



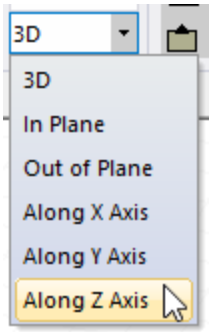
The next point you select will be on the same plane as the reference point in the positive or negative y direction.

Moving the Cursor Along the Z-Axis

To move the cursor to a point away from the reference point in the z direction:

1. Click the desired drawing command.
2. Do one of the following:

- Click **Modeler>Movement Mode>Along Z Axis**.
- Hold the shortcut key **Z**.
- Select **Along Z Axis** from the **Movement mode** pull-down list on the **Draw** ribbon.



The next point you select will be on the same plane as the reference point in the positive or negative z direction.

Choosing Snap Settings

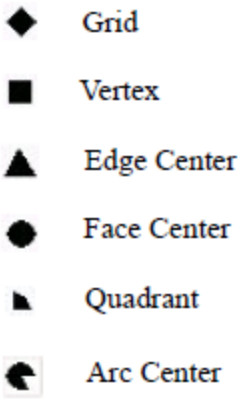
By default, the selection point and graphical objects are set to "snap to", or adhere to, a point on the grid when the cursor hovers over it. The coordinates of this point are used, rather than the exact location of the mouse. The cursor changes to the shape of the snap mode when it is being snapped.

To change the snap settings for the active design, you can use either the **Modeler** menu or the Draw ribbon Snap icons :

1. Click **Modeler>Snap Mode** or click the Draw ribbon **Snap** icons.

   If you select the menu command, the **Snap Mode** dialog box appears.
2. Specify the snap mode settings you want.
   - If you want the cursor to snap to a point on the grid, select **Grid** .
   - To snap to a vertex, select **Vertex**.
   - To snap to the center point of an edge, select **Edge Center** . The center point may be on a 1D, 2D, or 3D object edge.
   - To snap to the center of an object face, select **Face Center**.
   - To snap to the nearest quarter point on an edge, select **Quadrant**.
   - To snap to the center of an arc, select **Arc Center**.

     When the cursor snaps to a point, it changes to one of the following snap mode shapes:



Note	By default, the mouse is set to snap to the grid, a vertex, an edge center, a face center, and the nearest quadrant. To modify the default snap settings for the active design and all new designs, modify the selections on the **Snap** panel under the **3D Modeler** options.

Related Topics

Snap Setting Guidelines

Snap Setting Guidelines

For each object (3D, 2D, or 1D), you can define snapping points. Then, when moving in the Modeler window, if the cursor approaches a snap point (gets closer in terms of screen coordinates, not 3D coordinates), the snap point is highlighted. If you then click it, it becomes selected, even if that point is out of the plane you are working in (for example, even if the point has 3D coordinates when you are working in a 2D plane).

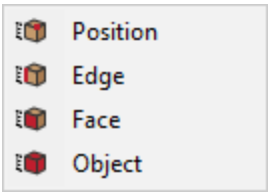
In general, select at least one of the snap options in the **Snap Mode** dialog box. If none of these options are selected, the software is in "free mode" and selects whatever point you click, regardless of its coordinates. This can cause problems when you are trying to create closed objects. Although the point you select may appear to be the vertex point of an open object, you may not have actually clicked the exact coordinates of the point.

Measure Modes for Objects

The Measure modes lets you measure the position, length, area, and volume of objects. With two faces selected, with two edges selected, or with an edge and a face selected, the Measure Mode displays the angle and distance between them. The Measure Position mode dynamically measures the distance between a reference point and the cursor location. You can Pause dynamic tracking in order to copy text information from the **Measure Data** dialog box, and then Resume dynamic updating.

1. To access the Measure mode, either:
   - Select **Modeler>Measure**.
   - Right-click and select **Measure** from the short-cut menu.

   After you select **Measure**, a cascading menu appears for **Position**, **Edge**, **Face** and **Object**. You can also select using the equivalent Draw ribbon icons:



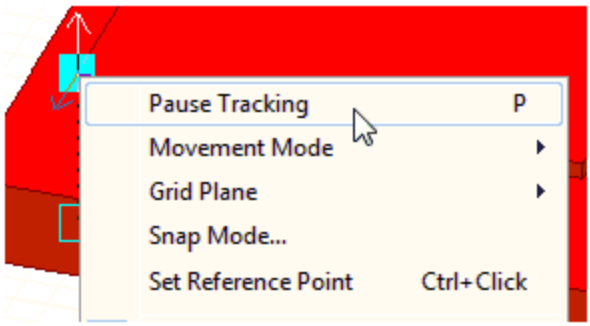
2. Select **Position** to obtain location and distance formation between a specified reference point and the cursor location.

   The **Measure Information** dialog box appears.

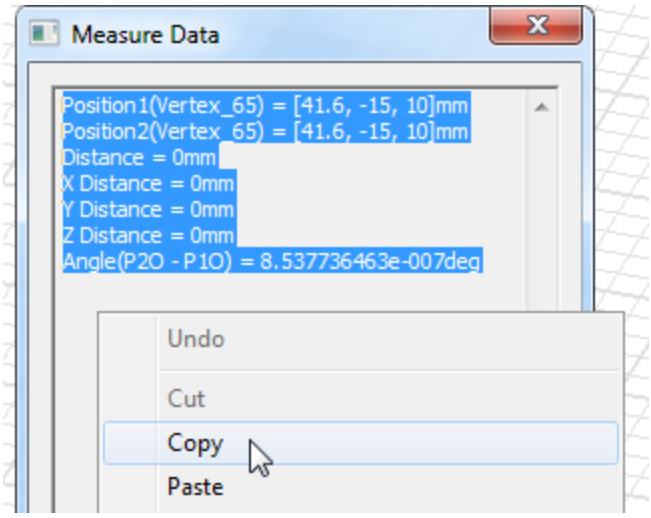
   With **Measure>Position** selected, the information displayed includes:
   - The location of the reference point. (Position1)
   - The current cursor location. (Position2)
   - The distance between the Reference and Current location.
   - The X distance.
   - The Y distance.
   - The Z distance.
   - The angle between the current reference point and the current cursor location.

Note	As you move the cursor, the Measure Information dialog displays the current cursor location and measurement information from the reference. Clicking on a new vertex updates the reference to the new location.

   To pause the dynamic tracking, either right-click and select **Pause**, or press P.



   While paused, you can move the cursor without changing the Measure Data. You can also cut and paste.



Press P or right-click and select **Resume** to resume cursor tracking.

With **Measure>Edge**, **Face**, or **Object** selected, the information displayed for each selected object is the name and:

- The area and volume of a 3D object.
- The area of a face.
- The length of a polyline (in edge selection mode, you can still see this if you select the polyline in the History tree)
- The length of an edge
- The location of a vertex.

For more information on cursor and reference point behavior in this mode, see Measuring Position and Distance

3. To use **Measure>Edge**, **Face**, or **Object** to measure the distance and angle between two selected items:
   - Select two points. Click the first and Ctrl-click to select the second.

     The **Measure Information** dialog displays the coordinates of each point, the distance between the points and the angle between Origin-P1, Origin-P2 line.
   - Select two faces, the **Measure Information** dialog displays the angle/distance between them.
   - The function is similar when you select two edges and when you select an edge and a face.
   - You can also measure distance between vertex/face, vertex/edge. In these cases, use the Select Multi mode.
4. To exit the Measure mode, click **Close** on the Measure Information dialog.

Related Topics

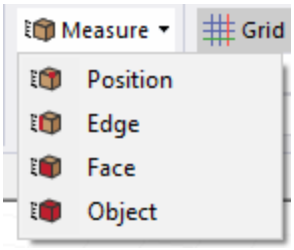
Measuring Position and Distance

Setting Coordinate Systems
Modifying the Coordinate System Axes View
Choosing Snap Settings
Choosing Movement Mode (3D, in plane, X, Y, or Z)
Select Multi (a mode for selecting objects, faces, edges or vertices)

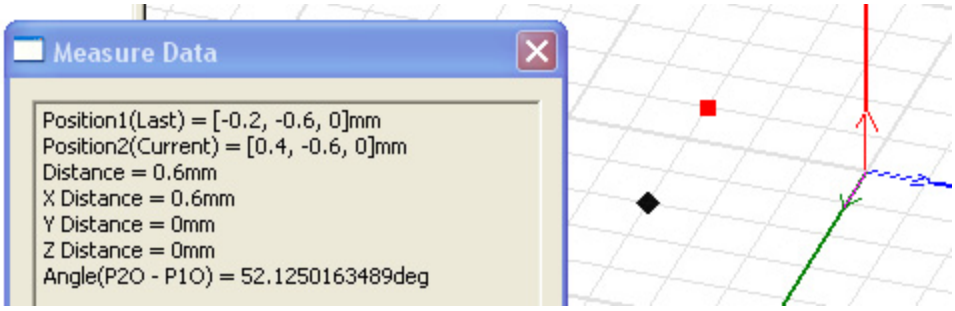
Measuring Position and Distance

To measure the distance between any cursor location relative to a designated reference point:

- Select **Modeler>Measure>Position** or use the equivalent icon on the **Draw** ribbon:

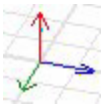


This enables the Measure Position mode and opens the **Measure Data** dialog box. The dialog lists the coordinates of the current reference point (Position1)and the cursor location (Position2). If you click, it shows the last position as a red square, and the current position as a black cursor. It also lists the distance between those points, the X, Y, and Z distances, and the angle between them. You can Pause dynamic tracking in order to copy text information from the **Measure Data** dialog box, and then Resume dynamic updating.



The shortcut menu displays the Hints item. When Hints are on (the default), a text display in the lower right of the 3D Modeler window, explaining how to set the reference point, and ways to control the movement mode.

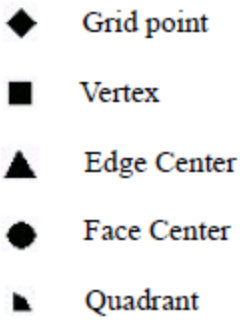
- The reference point is displayed as a mini x-y-z-axis:



Use Ctrl-Click to set the reference point at a new location.

- The cursor leads a diamond-shape selection marker that snaps from grid point to grid point. The **Measure Data** dialog also provides a text identification of the current grid points. If you drag the cursor off design objects, by default, it moves in the xy-plane. You can restrict movement to in a specific plane, out of plane, or z,x, or y. Besides the context menu for movement, you can also use the X, Y, and Z keys to restrict movement. See *Choosing a Movement Mode.* for further details.

  If you drag the selection marker over an object, it follows the 3D surfaces of the object, dropping a dashed reference line to a point on the current plane. The cursor changes shape to provide information about the object at the corresponding coordinate:

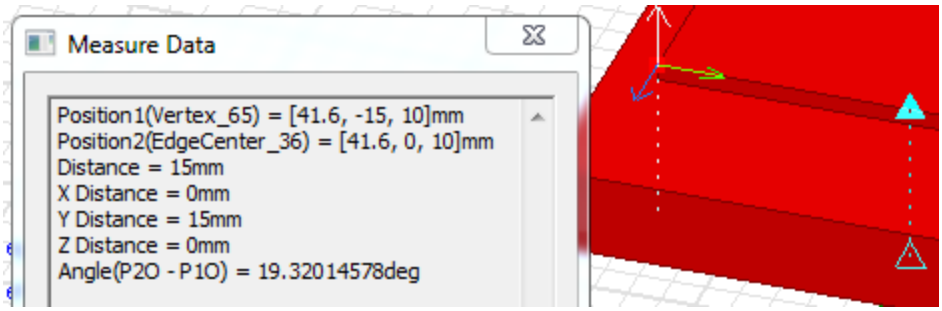


To measure the distance between two points:

1. Select **Modeler>Measure>Position** to enter Measure Position mode.
2. Ctrl-click to set the reference point.

   The reference point display moves to the selected point. This becomes the coordinate for Position1 in the Measure Data dialog box.
3. Drag the cursor to the second point.

   The value of the Position2 dynamically changes as you drag the cursor. You do not need to click. The values shown include:
   - Distance.
   - X distance
   - Y Distance.
   - Z Distance.
4. You can click P or right-click and select **Pause** from the shortcut menu to stop dynamic tracking. You can then copy text from the **Measure Data** dialog.



5. To resume dynamic tracking, either press P or right click and select Resume.
6. To close the dialog box and exit **Measure** mode, click the **Close** button. You can also use the ESC key to exit Measure mode.

Related Topics

Measure Modes for Objects

Setting Coordinate Systems

Modifying the Coordinate System Axes View

Choosing Snap Settings

Choosing Movement Mode (3D, in plane, X, Y, or Z)

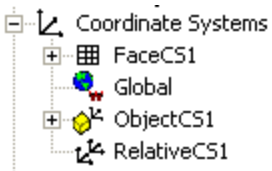
Select Multi (a mode for selecting objects, faces, edges or vertices)

Setting Coordinate Systems

The modeler has three types of coordinate systems that enable you to easily orient new objects:

- A *global* coordinate system.
- A *relative* coordinate system.
- A *face* coordinate system.
- An *object* coordinate system.

Every coordinate system (CS) has an x-axis that lies at a right angle to a y-axis, and a z-axis that is perpendicular to the xy plane. The origin (0,0,0) of every CS is located at the intersection of the x-, y-, and z-axes.The default Global coordinate system and any additional coordinate systems that you create for a project appear in the History tree of the modeler window.



- The *globalcoordinate system* (CS) is the fixed, default CS for each new project. It cannot be edited or deleted.
- A *relativeCS* is user-defined. Its origin and orientation can be set relative to an existing CS. Relative CSs enable you to easily draw objects that are located relative to other objects. If you modify a relative CS, all objects drawn on that CS will be affected and change position accordingly. You can define a relative CS to be *offset* and/or *rotated* from an existing CS. This feature provides a way for objects made of the same anisotropic materials to have different orientations.

  When you set a new relative coordinate system, you specify whether to express the coordinates as Absolute or Relative Coordinates. Absolute uses the specified values in terms of the global coordinate system. Relative interprets the values as differences from the current working CS.

  You have choices for expressing the coordinates as Cartesian, Cylindrical, or Spherical. These are evaluated as cartesian for the coordinate system properties.
- A *faceCS* is also user-defined. Its origin is specified on a planar object face. Face CSs enable you to easily draw objects that are located relative to an object's face.
- An *Object CS* is user-defined as attached to a specific object.

Switch between global, relative, object, and face CSs by changing the *workingCS*. Simply click the CS you want to use in the history tree. The working CS is indicated by a red *W* that appears at the lower-left corner of the CS name in the history tree. The **Properties** dialog box lists the CS associated with an object as the Orientation. By default, this is Global, but if you have created the object under a different coordinate system, that will be the orientation. You can click on the current orientation to see a drop down list of other orientation that you can assign for an object.

User-defined CSs are saved with the active project. When you open a project, it uses the CS designated as working CS when you last saved.

Related Topics

Creating a Relative Coordinate System

Creating a Face Coordinate System

Setting the Working Coordinate System

Modifying the Coordinate System Axes View

Change the Orientation of an object

Setting the Working Coordinate System

The working coordinate system is the current CS with which objects being drawn are associated. The working CS can be the global CS or a user-defined relative CS or face CS. Select the working CS by clicking its name in the history tree, or do the following:

1. Click **Modeler>Coordinate System>Set Working CS**.

   The **Select Coordinate System** dialog box appears.

2. Click a CS in the list.
3. Click **Select**.

   A red ***W*** appears at the lower-left corner of the CS name in the history tree, indicating that it is the working CS.

   Objects that you draw from now on are drawn in and associated with the CS you selected.

Related Topics

Setting Coordinate Systems

Creating a Relative Coordinate System

When creating a relative CS, you have the following options:

- You can create an offset relative CS, that is, a relative CS whose origin lies a specified distance from another CS's origin. By moving a CS's origin, you can enter coordinates relative to an existing object, without having to add or subtract the existing object's coordinates.
- You can create a rotated relative CS, that is, a relative CS whose axes are rotated away from another CS's axes. By rotating the axes of a CS, you can easily add an object that is turned at an angle relative to another object.
- You can also create a relative CS that is both offset and rotated.

Creating an Offset Relative CS

To create a relative CS with an origin that lies a specified distance from another CS's origin:

1. In the history tree, click the CS upon which you want to base the new relative CS, making it the working CS.
2. Click **Modeler>Coordinate System>Create>Relative CS>Offset**.
3. On the **Relative CS** menu, click **Offset**  .
4. Select the origin in one of the following ways:
   - Click the point.
   - At the lower right of the modeler window, use the drop down menu to select the system for expressing coordinates (Cartesian, Cylindrical, or Spherical), select either relative or absolute coordinates, then select the units, and type the CS origin coordinates in boxes.

   To select a point that does not lie in the current plane, use the **Movement Mode** commands on the shortcut menu.

   The new relative CS is created. Its origin has moved from the previous working CS, but its axes remain the same. It is listed in the history tree under **Coordinate Systems**. It automatically becomes the working CS; objects that you draw hereafter will be based on the coordinates of this relative CS. Default planes are created on its xy, yz, and xz planes.

Related Topics

Creating a Relative Coordinate System
Creating an Offset and Rotated Relative CS

Creating a Rotated Relative CS

To create a new relative CS with its axes rotated away from another CS's axes:

1. In the history tree, select the CS upon which you want to base the new relative CS, making it the working CS.
2. Click **Modeler>Coordinate System>Create>Relative CS**>**Rotated**  .
3. Specify the x-axis by selecting a point in one of the following ways:
   - Click the point.
   - At the lower right of the modeler window, use the drop down menu to select the system for expressing coordinates (Cartesian, Cylindrical, or Spherical), select either relative or absolute coordinates, then select the units, and type the CS origin coordinates in boxes.

   To select a point that does not lie in the current plane, use the **Movement Mode** commands on the shortcut menu.
4. Specify the xy plane by selecting any point on it in one of the following ways:
   - Click the point.
   - Type the coordinates of a point that is relative to the previously selected point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   You do not need to specify the z-axis. It is automatically calculated so that the axes form a right-handed Cartesian coordinate system.

   The new relative CS is created. It has the same origin as the previous working CS, but its axes are rotated. It is listed in the history tree under **Coordinate Systems**. It automatically becomes the working CS; objects that you draw hereafter are based on the coordinates of this relative CS. Default planes are created on its xy, yz, and xz planes.

Related Topics

Creating a Relative Coordinate System
Creating an Offset and Rotated Relative CS

Creating an Offset and Rotated Relative CS

To create a new relative CS that is both offset and rotated from an existing CS:

1. In the history tree, select the CS upon which you want to base the new relative CS, making it the working CS.
2. Point to **Modeler>Coordinate System>Create>Relative CS**.

3. On the **Relative CS** menu, click **Both**  .
4. Select the origin in one of the following ways:
   - Click the point.
   - At the lower right of the modeler window, use the drop down menu to select the system for expressing coordinates (Cartesian, Cylindrical, or Spherical), select either relative or absolute coordinates, then select the units, and type the CS origin coordinates in boxes.

   To select a point that does not lie in the current plane, use the **Movement Mode** commands on the shortcut menu.
5. Specify the x-axis by selecting a point on the axis in one of the following ways:
   - Click the point.
   - Type the coordinates of a point that is relative to the origin in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.
6. Specify the xy plane by selecting any point on it in one of the following ways:
   - Click the point.
   - Type the coordinates of a point that is relative to the previously selected point in the **dX**, **dY**, and **dZ** boxes.

   You do not need to specify the z-axis. It is automatically calculated so that it is at a right angle to the y-axis.

   The new relative CS is created. It is listed in the history tree under **Coordinate Systems**. It automatically becomes the working CS; objects that you draw hereafter are based on the coordinates of this relative CS. Default planes are created on its xy, yz, and xz planes.

Related Topics

Creating a Relative Coordinate System

Creating a Face Coordinate System

1. Select the object face upon which you want to create the face CS.
2. Click **Modeler>Coordinate System>Create>Face CS**  .
3. Select the origin in one of the following ways:
   - Click the point on the face.
   - Type the point's coordinates in the **X**, **Y**, and **Z** boxes.
4. Specify the x-axis by selecting a point on the object face in one of the following ways:
   - Click the point.
   - Type the coordinates of a point that is relative to the previously selected point in the **dX**, **dY**, and **dZ** boxes, where **d** is the distance from the previously selected point.

   You do not need to specify the y- or z-axes. The modeler assumes that the z-axis is normal to the object face and the y-axis is automatically calculated to be at a right angle to the z-axis.

The new face CS is listed in the history tree under **Coordinate Systems**. It automatically becomes the working CS; objects that you draw hereafter are referenced to the coordinates of this face CS. Default planes are created on its xy, yz, and xz planes.

Only operations listed in the history tree *before* the face CS's creation affect the face CS, and in turn, affect objects dependent upon that face CS. A face CS, or objects created on it, are *not* affected by operations that occur after it is created.

For example, suppose you create a box, then a face CS on a face of the box, and then a cylinder on the face CS. If you then edit the box's dimensions in the **Properties** dialog box, the cylinder moves accordingly. But if you rotate the box using the **Edit>Arrange>Rotate** command, the box moves, but the cylinder does not because the operation occurs later in the history tree.

Related Topics

Automatically Creating Face Coordinate Systems

Setting the Working Coordinate System

Modifying Coordinate Systems

Setting Coordinate Systems

Automatically Creating Face Coordinate Systems

You can instruct Maxwell to automatically create a new face CS every time you draw on an object's face.

1. Click **Tools>Options>Modeler Options**.

   The **Modeler Options** dialog box appears.
2. In the **Operation** tab, select **Automatically switch to face coordinate system**.
3. Click **OK**.

Now, when you select a face, and then click any drawing command, a new face CS is created. The modeler automatically sets the new face CS as the working CS. The object you draw is oriented according to the new face CS.

Note	The modeler does not automatically create a new face CS if a face CS has already been assigned to the selected face.

Related Topics

Creating a Face Coordinate System

Creating an Object Coordinate System

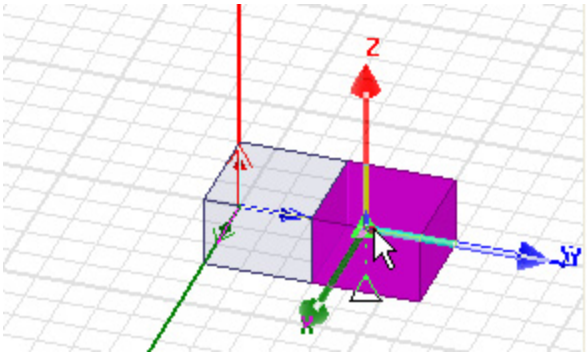
You can create coordinate systems based on any object of solid, sheet, or wire type. The **Modeler>Coordinate System>Create>Object CS** command is enabled when you select an

object. An Object CS can be Offset, Rotated or both. Executing one of the Object CS commands changes the cursor to the selection marker mode.

As you drag the selection marker over an object, it follows the 3D surfaces of the object, dropping a dashed reference line to a point on the current plane. The cursor changes shape to provide information about the object at the corresponding coordinate:

- Grid point
- Vertex
- Edge Center
- Face Center
- Quadrant

For example, in this case, the cursor shows Edge Center triangles as valid selection points for an **Modeler>Coordinate Systems>Create Object CS>Offset** command.



Only operations listed in the history tree *before* the Object CS's creation will affect the Object CS, and in turn, affect objects dependent upon that Object CS. An Object CS, or objects created on it, is *not* affected by operations that occur after it is created. Also see the Move CS to End command.

For example, suppose you create a box, then an Object CS on a face of the box, and then a cylinder on the Object CS. If you then edit the box's dimensions in the **Properties** dialog box, the cylinder will move accordingly. But if you rotate the box using the **Edit>Arrange>Rotate** command, the box will move, but the cylinder will not move because the operation occurs later in the history tree.

Related Topics

Creating a Face Coordinate System

Move CS to End

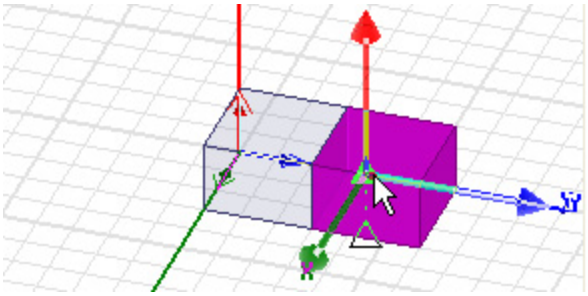
Creating an Offset Object CS

Creating a Rotated Object CS
Creating an Object CS that is both Rotated and Offset
Expressing Cartesian, Cylindrical, or Spherical Coordinates

Creating an Offset Object CS

To create an Offset Object coordinate system (CS):

1. Select the working CS.
2. Select the object, and click **Modeler>Coordinate System>Create>Object>Offset**.
   - You can select any snap point on object based on current snapping mode to select origin of the CS. When you hover the mouse over a valid point, a coordinate system preview is shown.



   - The point must be on selected object.
   - X axis is taken as {1,0,0} and Y Axis as {0,1,0} to create an Object CS. The axis coordinates can be later edited through the Properties dialog.
   - When you select the point, validation displays an appropriate message. Points where CS preview is available are always valid.

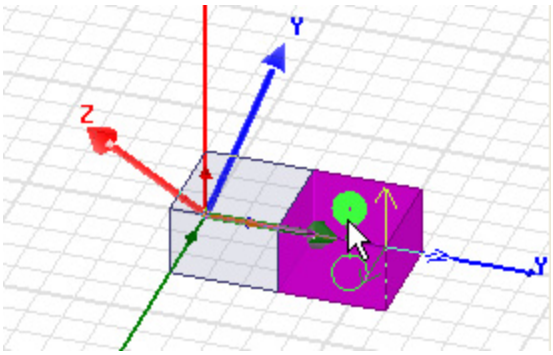
Related Topics

Creating a Face Coordinate System
Move CS to End
Creating an Offset Object CS
Creating a Rotated Object CS
Creating an Object CS that is both Rotated and Offset
Expressing Cartesian, Cylindrical, or Spherical Coordinates

Creating a Rotated Object CS

To create a Rotated Object coordinate system (CS):

1. Select the working CS.
2. Select the object, and click **Modeler>Coordinate System>Create>Object>Rotated**
   - The origin is taken as {0,0,0}. You can edit this later through the Properties window.
   - The GUI is in multi select mode where you define the direction by picking any of vertex, straight edge, planar face or conical face.
     - If you select vertex, direction from origin to vertex defines the axis.
     - If you select a straight edge, the edge direction defines the axis.
     - If you select a planar face, face normal defines the axis.
     - If you select a conical face, the face axis defines the CS axis.
   - The selected vertex, edge or face must be on the selected object.
3. You are prompted to first select X axis. A preview of X axis is shown for valid selections.
4. You are later prompted to select to define the XY plane. Another direction (edge or face selection) or point (i.e. vertex selection) helps to define the plane.
   - During 2nd XY plane selection, a CS preview shows as the cursor hovers over valid selections. The preview includes the three CS axes and the XY plane.



   - In 2D modeler, you are prompted to select only X axis. The Y axis is defined based on 2D modeler type. This resembles **Relative CS** behavior. Face picking is not available in 2D modeler as it defines the direction out of plane.

Related Topics

Creating a Face Coordinate System

Move CS to End

Creating an Offset Object CS

Creating a Rotated Object CS

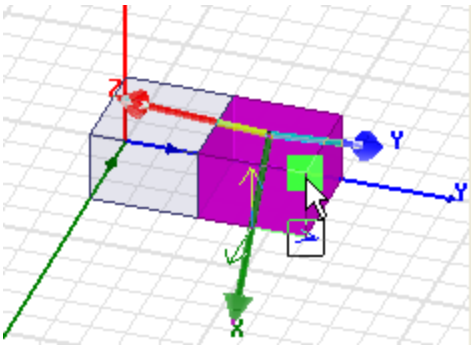
Creating an Object CS that is both Rotated and Offset

Expressing Cartesian, Cylindrical, or Spherical Coordinates

Creating an Object CS that is Both Offset and Rotated

To create an Object CS that is both offset and rotated:

1. Set the working CS.
2. Select the object
   - Behavior is combination of offset and rotated object CS. You are first prompted to select origin as in **Object CS>**Object and then axes as in **Object CS>**Rotated.



   - In 2D modeler, you are asked to select only the origin and X axis. The Y axis is computed as for **Object CS>Rotated**. Face picking is not available in 2D modeler as it defines the direction out of plane.

Related Topics

Creating a Face Coordinate System

Move CS to End

Creating an Offset Object CS

Creating a Rotated Object CS

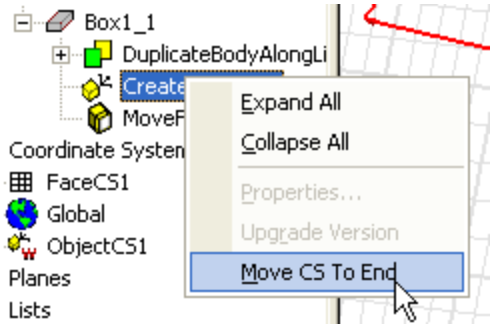
Creating an Object CS that is both Rotated and Offset

Expressing Cartesian, Cylindrical, or Spherical Coordinates

Move CS to End Command for History Tree

Only operations listed in the history tree *before* the Object CS's creation will affect the Object CS, and in turn, affect objects dependent upon that Object CS. An Object CS, or objects created on it, is *not* affected by operations that occur after it is created. It is sometimes useful to have the coordinate system affected after any other operations that might have edited, moved or rotated the object.

If you have at least one History operation after a Face or Object CS in the History tree, selecting a Face or Object CS enables the **Move CS to End** command in the **Modeler>Coordinate System** cascade menu and the History Tree shortcut menu.



Executing the command moves the selected CreateObject CS to the end position in the History tree and updates associated items (other CS, object history, any dependent parts etc.).

Related Topics

Creating a Face Coordinate System

Move CS to End

Creating an Offset Object CS

Creating a Rotated Object CS

Creating an Object CS that is both Rotated and Offset

Modifying Coordinate Systems

Keep in mind that when you edit a CS, the following are also affected:

- All objects drawn in the CS.
- All CSs that were defined relative to that CS.
- All objects drawn on a CS that was defined relative to that CS.

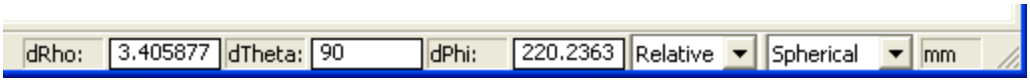
There are two ways to modify a coordinate system: you can select the coordinate system in the history tree in the modeler window, and open its properties dialog box. This approach does not also allow you to change whether the coordinate system is Absolute or Relative, or to change how you express the coordinates (as Cartesian, Cylindrical, or Spherical).

If you want to also modify the whether the coordinate system is Absolute or Relative, and to change how your express the coordinate, do the following:

1. Click **Modeler>Coordinate System>Edit**.

   The **Select Coordinate System** dialog box appears.
2. Click the CS you want to modify.
3. Click **Select**.

This selects that coordinate system and enables the editable fields at the lower right of the Modeler window. After you click the cursor in the first field, you can type in values, and tab to the next fields.



4. You can select Absolute or Relative as the Coordinate system If you selected a relative CS, follow the directions for creating a relative CS.

   If you selected a face CS, follow the directions for creating a face CS.
5. You select the coordinate system from the drop down menus as Cartesian, Cylindrical, or Spherical.
6. Select the units from the drop down menu.

   The value you give here is translated to Cartesian coordinates in the Properties for the Coordinate system.

Related Topics

Setting Coordinate Systems

Creating a Relative Coordinate System

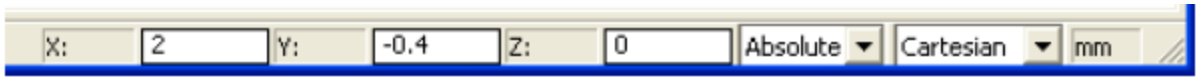
Creating a Face Coordinate System

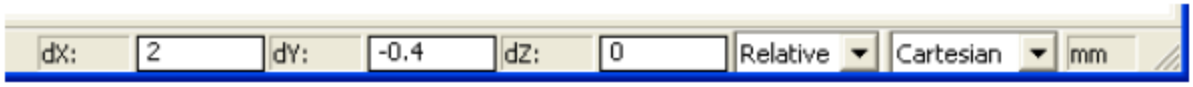
Modifying the Coordinate System Axes View

Expressing Cartesian, Cylindrical, or Spherical Coordinates

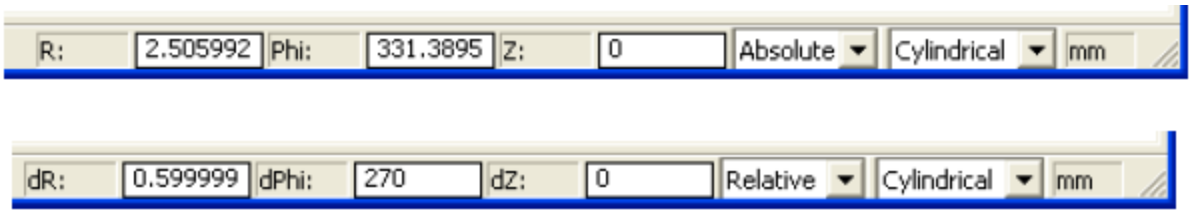
You have choices for expressing the coordinates for as Cartesian, Cylindrical, or Spherical. In each case, you also specify whether to enter the coordinates as Absolute or Relative to the working coordinate system. Click the cursor in the first text field to begin entering values or variables. Tab to the next fields, and Enter when your are done. After you enter values or variables in the text fields, they are evaluated as Cartesian for the Properties window for that coordinate system.

- **Cartesian**, that is the point's distance from the origin in the x, y, and z directions in the **X**, **Y**, and **Z** text boxes.

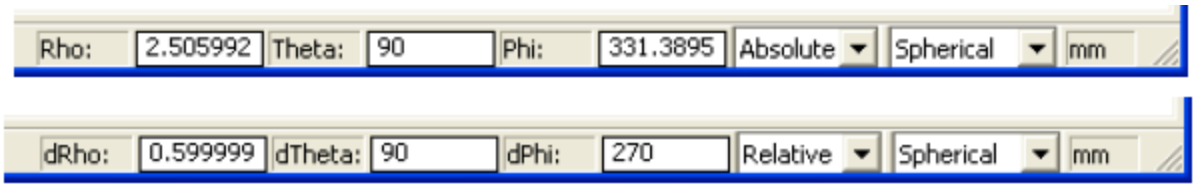




- **Cylindrical**, that is, the point's radius, measured from the origin, in the **R** text box, the angle from the x-axis in the **Theta** text box, and the distance from the origin in the z direction in the **Z** text box.



- **Spherical**, that is, in the point's radius, measured from the origin, in the **Rho** text box, the angle from the x-axis in the **Theta** text box, and the angle from the origin in the z direction in the **Phi** text box.



Related Topics

Defining Absolute Coordinates
Defining Relative Coordinates
Defining Cartesian Coordinates
Defining Cylindrical Coordinates
Defining Spherical Coordinates

Deleting Coordinate Systems

1. Click the name of the CS you want to delete in the history tree.
2. Click **Edit>Delete**  .
   - Alternatively, press **Delete**.

   The CS is deleted and all objects drawn on it are deleted. Further, any CS that depended on the deleted CS is deleted and any objects that were drawn in the dependent CS are also be deleted.

Related Topics

Setting Coordinate Systems
Creating a Relative Coordinate System

Creating a Face Coordinate System
Modifying the Coordinate System Axes View

Setting the Drawing Plane

The **Drawing plane** pull-down list is available on the **Draw** ribbon.

To set the drawing plane, do one of the following:

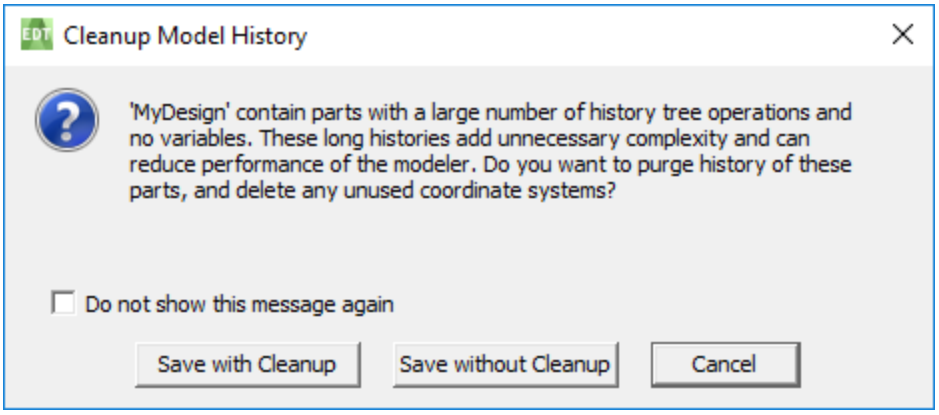
- Select **XY** from the **Drawing plane** pull-down list on the **Model** ribbon.
- Click **Modeler>Grid Plane**, and then select a grid plane: **XY**, **YZ**, or **XZ**.

Cleaning Up Model History

Ansys Electronics Desktop can clean up history tree operations in order to improve performance.

When saving your project, you will be prompted to clean up your model's history if some parts meet two conditions:

- No variables assigned to the part
- A history greater than a specified number of actions



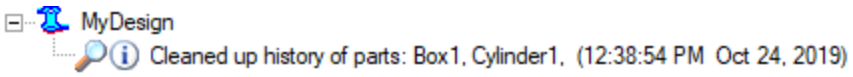
The number of actions that triggers this prompt can be set in 3D Modeler Operations Options.

When promoted, you can choose **Do not show this message again** to hide the prompt for the active project.

To clean up the model history manually:

- Select **Modeler** > **Cleanup Model History**.

After the model has been cleaned up, you can view cleaned up parts in the **Messages** window:



Related Topics:

3D Modeler Options: Operation Options

9 - Maxwell User Defined Primitives

User Defined Primitives (UDPs) are compiled libraries of commonly used geometric models in electromechanical energy devices and power systems that can be added to the desktop interface and shared between users with common modeling needs. UDPs allow you to add customized geometric models to the Maxwell Desktop.

- In Maxwell, many commonly used UDPs have been defined, especially for RMxprt UDPs. For more information, see RMxprt UDPs and Helix UDPs.
- You can also define your own UDPs by writing source code and then generating .dll files for UDPs. For more information, see Source Code Files.

Launching a UDP from Maxwell Desktop

In the Maxwell desktop, click **Draw>User Defined Primitive** to choose a UDP. For example, a band UDP is launched as follows:

1. Click **Draw>User Defined Primitive>RMxprt>Band**, as shown in the following figure**.**

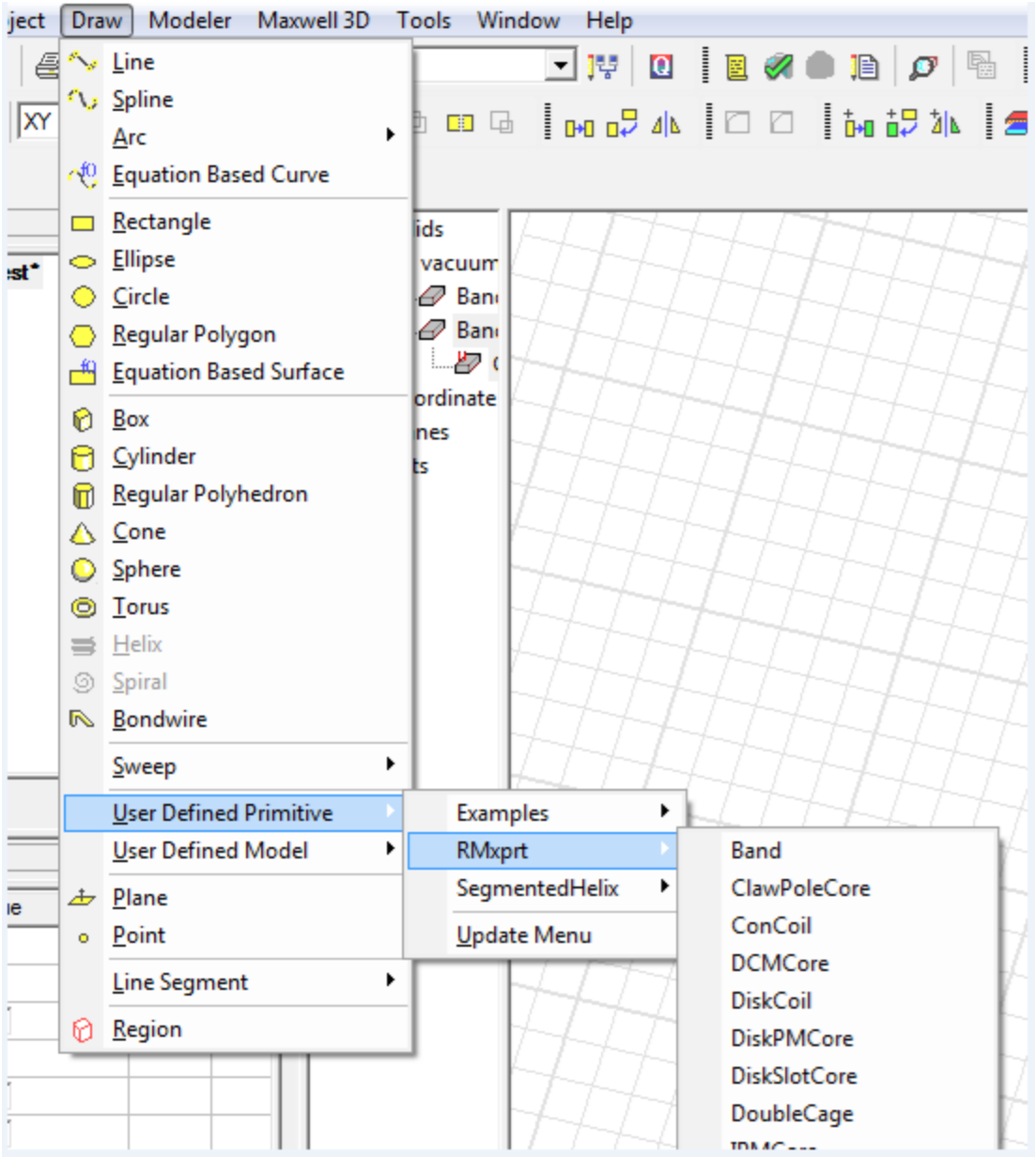


Figure 9-1 Create a band

A **User Defined Primitive Operation** dialog opens.

On the **Parameters** tab, you can view and edit the parameters. An **Info** tab contains information about the user defined primitive, its purpose, the company/author who created it, the date created and the version number.

2. Assign the values in the **Parameters** tab, and then click **OK** to finish setting parameters.

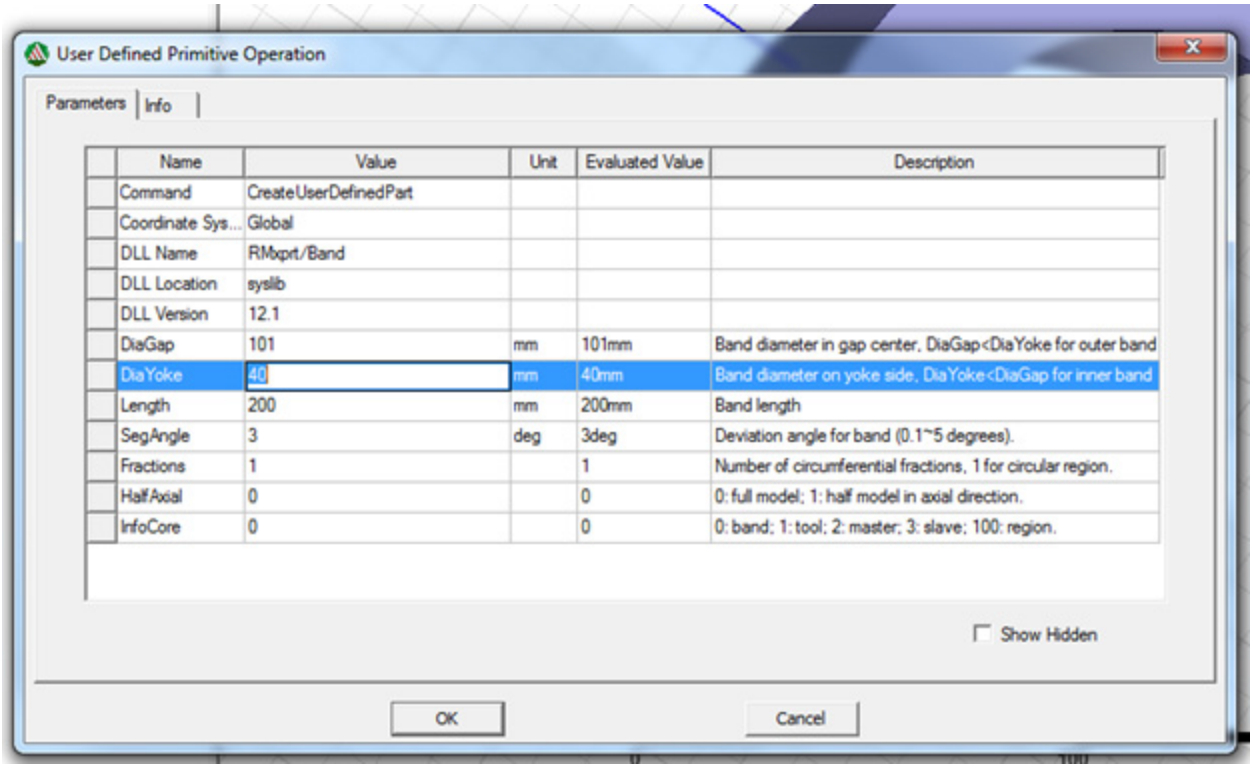


Figure 9-2 Assign parameters values

A UDP model appears in the coordinate system window.

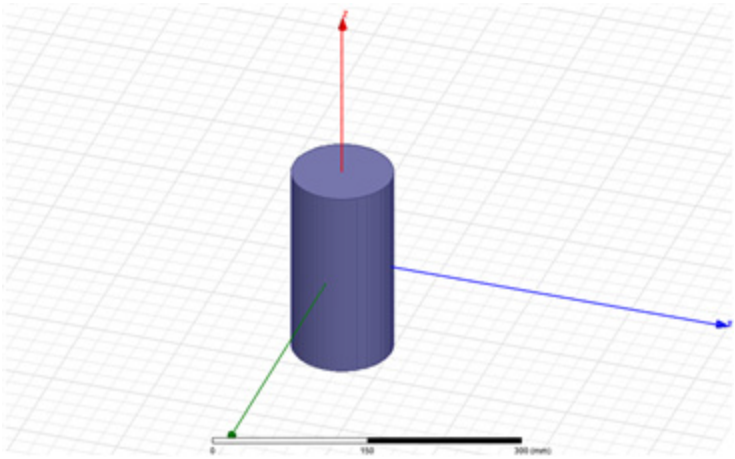


Figure 9-3 A band UDP is created

RMxprt UDPs

RMxprt UDPs are 2D and 3D geometries created for finite element analysis of electrical machines.

RMxprt UDPs consist of the following types:

- UDPs for Band and Regions
- UDPs for Slot Cores and Coils
- UDPs for Pole Cores and Coils
- UDPs for Other Cores and Coils
- UDPs for Press Plates, Fingers, and Stems

UDPs for Bands and Regions

RMxprt includes two types of UDPs for defining bands and regions:

- Band UDP
- Core regions

Band UDP

The Band UDP is used to create:

- A region to include all moving objects, which is called a “band”
- An outer region to include all moving and static objects
- A region used as a subtract tool to derive a partial model
- Independent and dependent boundaries

Property	Description
DiaGap	Band diameter in gap center, DiaGap < DiaYoke for outer band
DiaYoke	Band diameter on yoke side, DiaYoke < DiaGap for inner band
Length	Band Length
SegAngle	Angle per segment of band (0.1 ~ 5 degrees)
Fractions	Number of circumferential fractions, 1 for circular region
HalfAxial	0: full model; 1: half model in axial direction.
InfoCore	0: band; 1: tool; 2: independent; 3: dependent; 100: region

These parameters are used in the following 2D inner band and 2D outer band examples:

Figure 9-4 Inner band

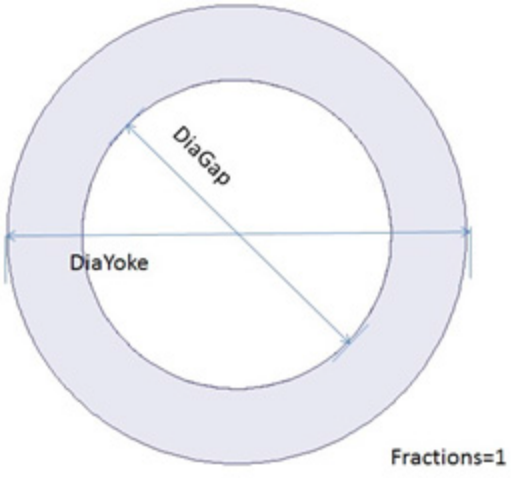


Figure 9-5 Outer band

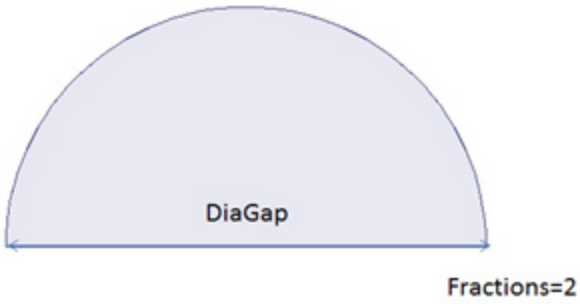


Figure 9-6 Inner band half

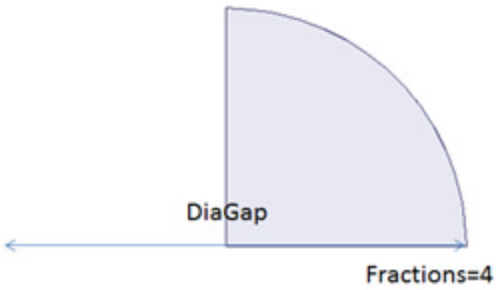


Figure 9-7 Inner band semi

Parameters such as **SegAngle**, **Length** and **HalfAxial** are shown in the following 3D figures:

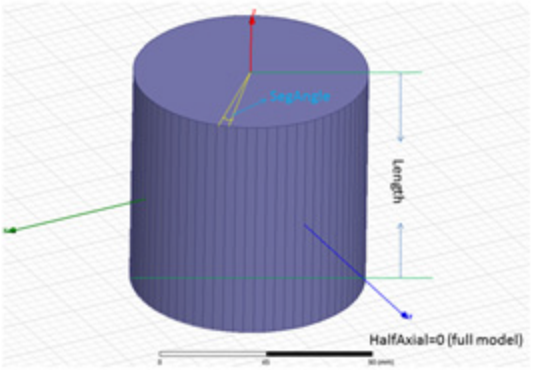


Figure 9-8 Inner band (full model)

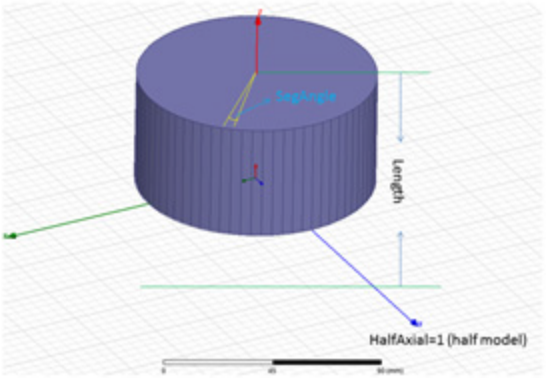


Figure 9-9 Inner band (half model)

Creating a Band

Maxwell provides you with a .dll file to define the parameters of a band.

1. Click **Draw>User Defined Primitive>RMxprt>Band**.
   The **User Defined Primitive Operation** dialog box appears.

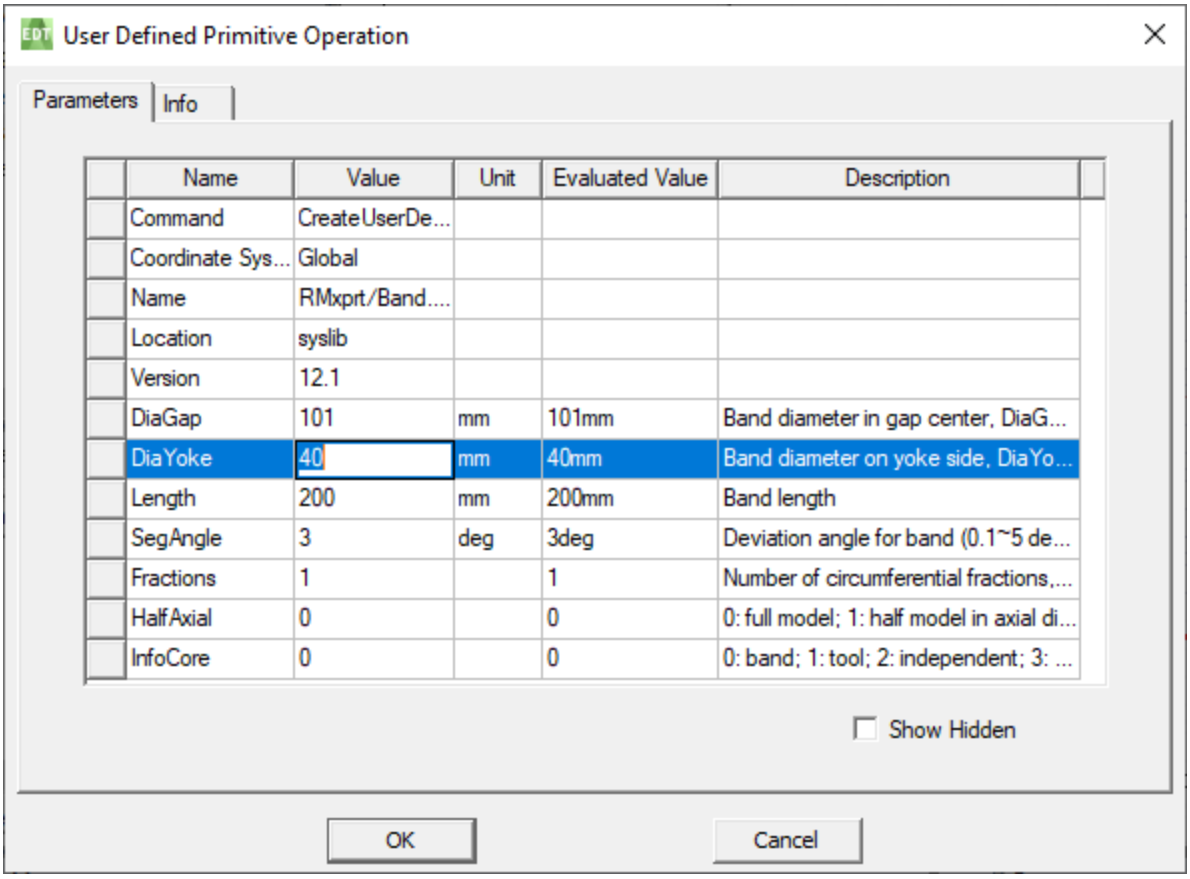


Figure 9-10 Change values of parameters when creating a UDP

The **Parameters** tab permits you to see and edit the parameters. An **Info** tab contains information about the user defined primitive, its purpose, the company/author who created it, the date created and the version number.

2. Specify the values for the parameters.
3. Click **OK**.

Creating an Outer Region

An outer region can be assigned by setting the **InfoCore** property to 100 either when creating the band or after creating it. The method for assigning during band creation is discussed in Creating a Band. The method for assigning after the band is created is as follows:

1. For assigning an outer region after the band is created, click **CreateUserDefinedPart** in the history tree under the specific band, as shown in Figure 9-11 . You can change the **InfoCore** property to **100** under **Command** tab.

2. For an outer region, set **DiaGap** < **DiaYoke**. For a shaft, set **DiaGap** > **DiaYoke**.

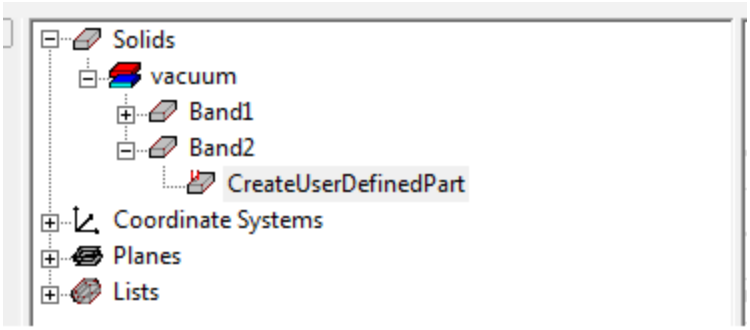


Figure 9-11 CreateUserDefinedPart in the history tree

Creating a Tool

When you are defining parameters, the **InfoCore** property can be assigned to **1**, which means the band is used as a tool. This property can be assigned either when creating the band or after creating it. The method for assigning during band creation is discussed in Creating a Band. For assigning after the band is created, click **CreateUserDefinedPart** in the history tree under the specific band, as shown in Figure 9-11 . You can then change the **InfoCore** property to **1** under **Command** tab.

Note	To be assigned as a tool, the band must be an outer band, which means the band's DiaGap value must be less than DiaYoke value.

A tool is mainly used for subtraction in the model.

For example, if a half model of NonSalientPoleCore UDP is needed, a tool is necessary to be used to subtract half of NonSalientPoleCore.

First, create a tool.

1. Click **Draw>User Defined Primitive>RMxprt>Band**.
2. Change **DiaGap** and **DiaYoke** values to set the **DiaGap** value to be less than the **DiaYoke** value.
3. Set **InfoCore** value to 1 and change the **Color** of the tool to another one in order to

distinguish the tool from the NonSalientPoleCore, which is created later.

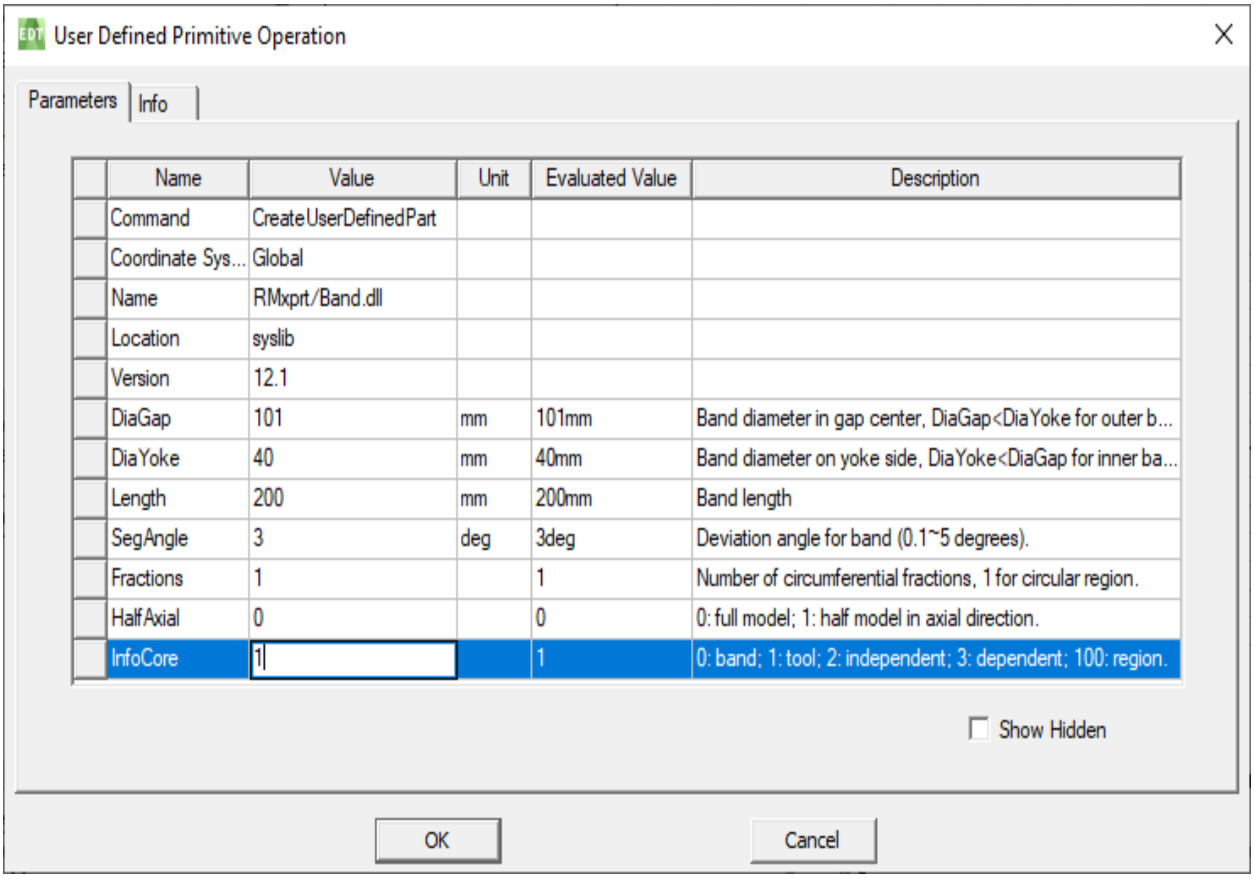


Figure 9-12 Set the values of parameters to create a tool

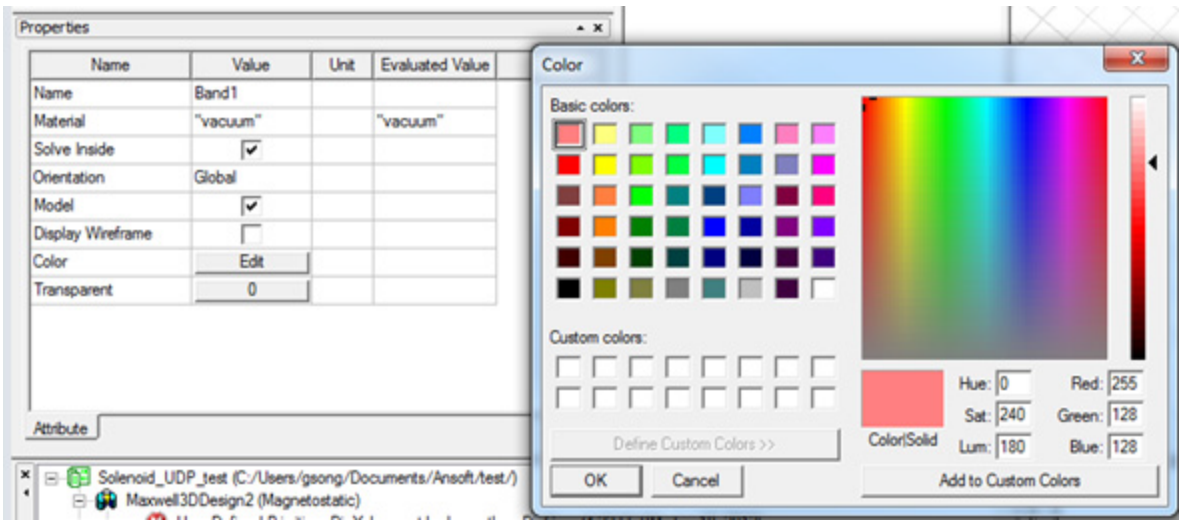


Figure 9-13 Change the color of the tool

Then, create a non-salient pole core.

1. Click **Draw>User Defined Primitive>RMxprt>NonSalientPoleCore**.
2. Click **OK** to finish setting the parameter.

The coordinate system window looks similar to the following figure:

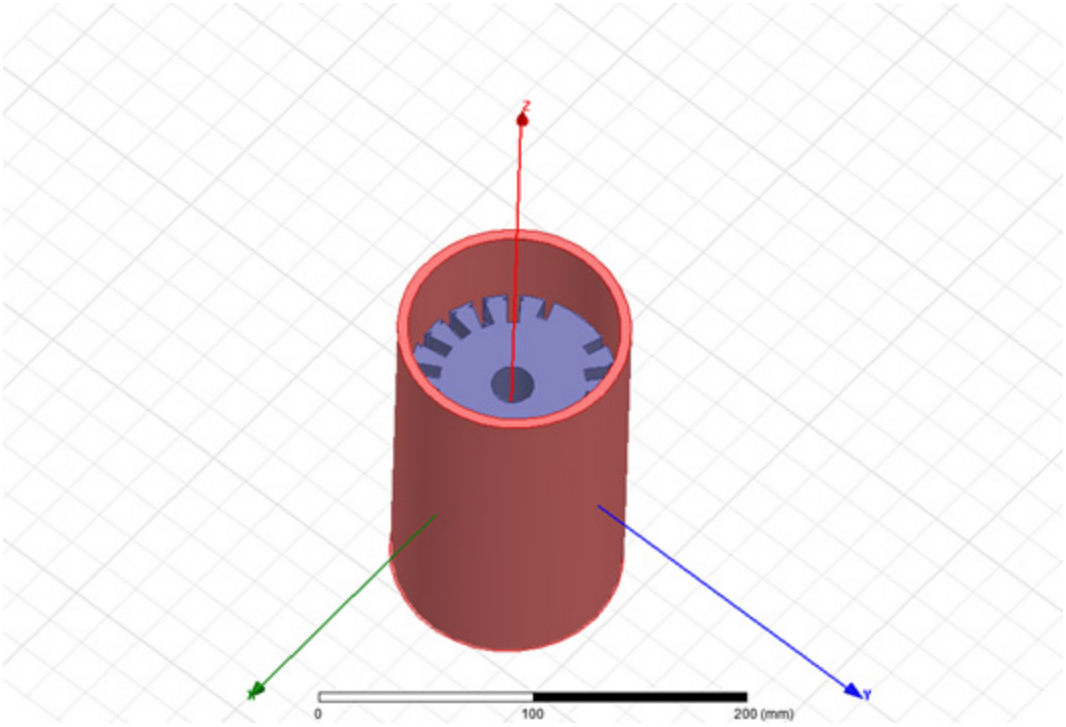


Figure 9-14 Non-Salient Pole Core (gray) with a tool

Change the **Fractions** of the tool to 2.

1. Click **CreateUserDefinedPart** in the history tree under the tool (named Band1 in the figure).
2. Change the value of **Fractions** to 2.

3. Click **OK** to finish setting the parameter.

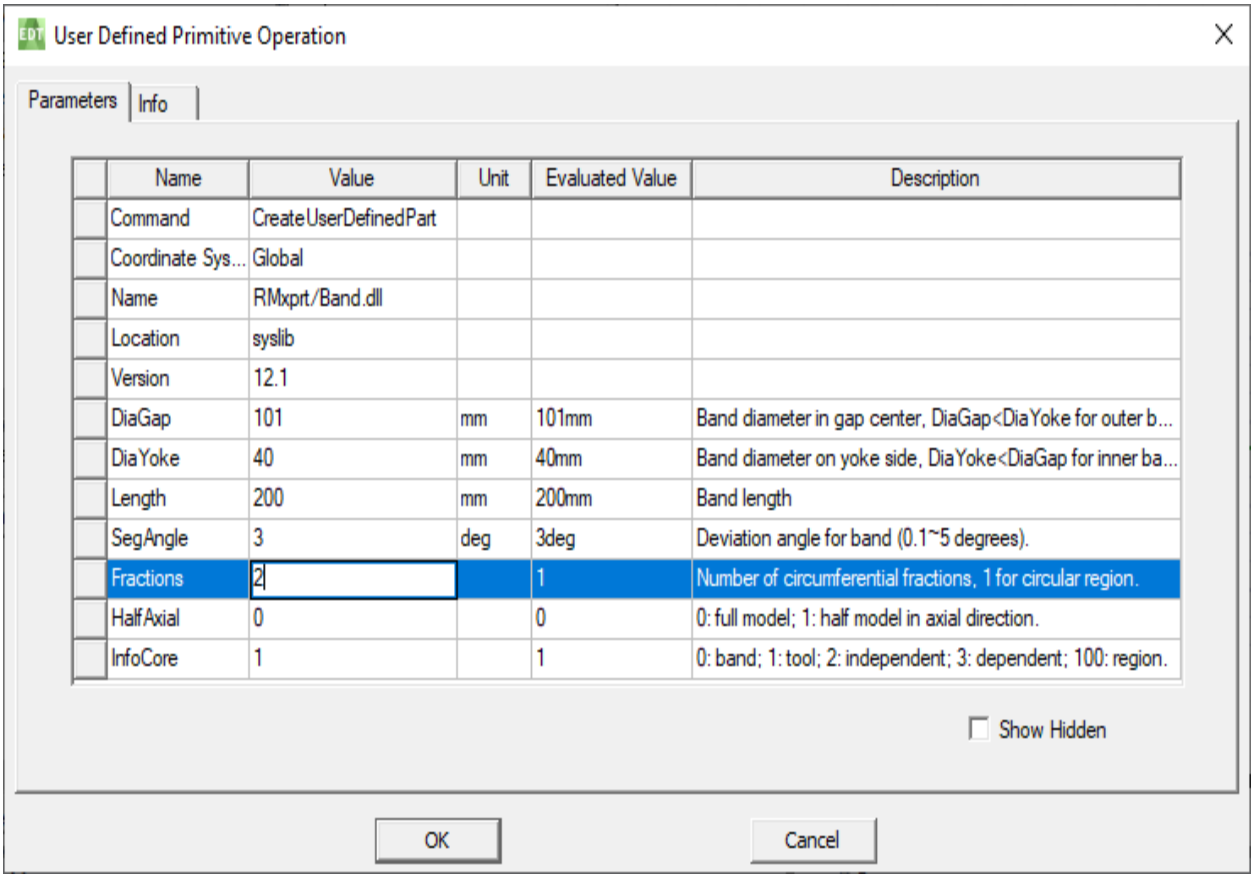


Figure 9-15 Change the value of Fractions of Non-Salient Pole Core (gray)

The last step is to subtract the non-salient pole core using the tool.

1. Choose NonSalientPoleCore1 and Band1 (tool) simultaneously by pressing **Ctrl** on the keyboard.

2. Click **Modeler>Boolean>Subtract** to do the subtraction.

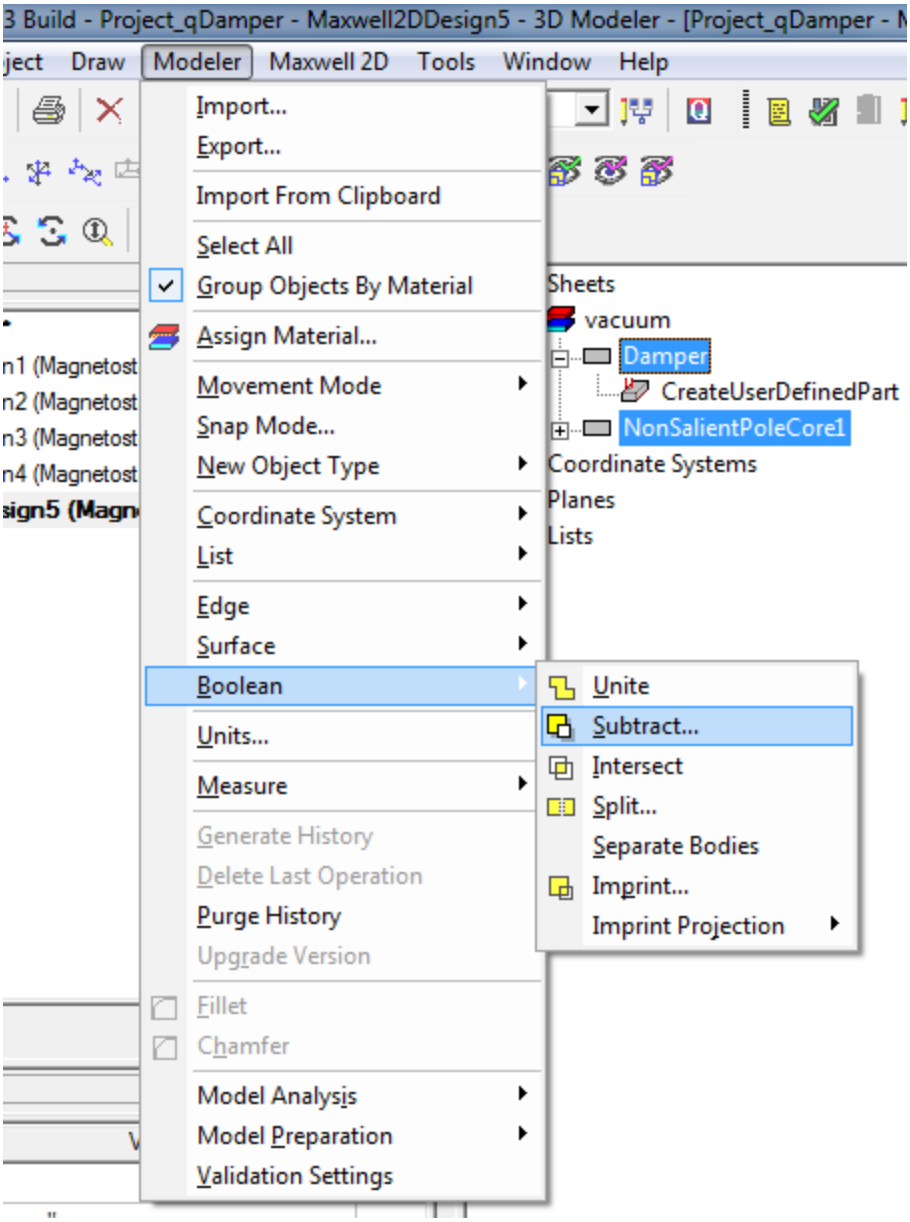


Figure 9-16 Subtraction of two objects

A Subtract window appears.

3. Choose NonSalientPoleCore1 as the **Blank Parts** and Band1 as the **Tool Parts**.
4. Click **OK** to finish subtraction.

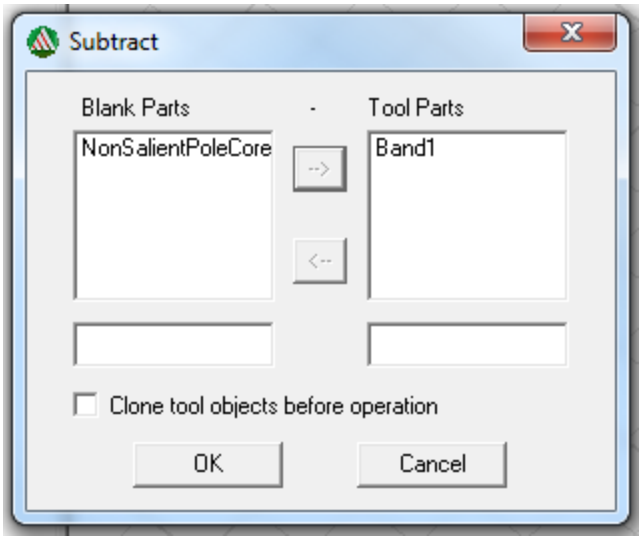


Figure 9-17 Set Blank Parts and Tool Parts

The final Non-Salient Pole Core is shown in the coordinate system window in the following figure:

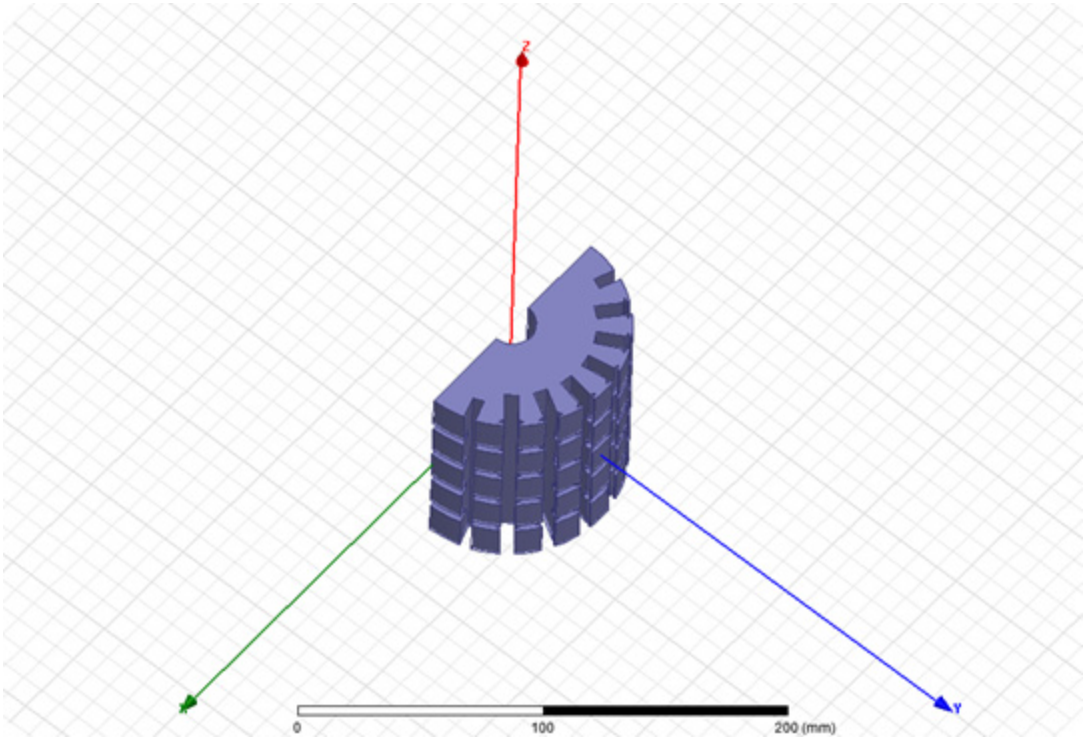


Figure 9-18 A non-salient pole core after subtraction

Creating 3D Independent and Dependent Boundaries

There are two ways to create 3D independent and dependent boundaries:

- Creating a new band and changing the entire band to an independent or dependent boundary
- Selecting one of the planar faces of an existing band in the coordinate window and assigning the face to an independent or dependent boundary

The process of changing an entire band to an independent or dependent boundary is similar to creating a tool. When creating a new band or after creation of a band, you can assign the **InfoCore** property to 2 (for independent) or 3 (for dependent). After clicking **OK**, the band is changed to a face.

The process of assigning independent and dependent boundaries on an existing face follows:

1. Click **CreateUserDefinedPart** in the history tree under the band for which you are assigning boundaries. Set the **Fractions** property of the band to an integer larger than 1 to make the band have a planar face.
2. Click **Edit>Selection Mode>Faces** or right-click anywhere in the coordinate window to choose **Selection Mode>Faces**.
3. Select a planar face of the band by right-clicking the face. Then select **Assign Boundary>Matching>Independent**(or **Assign Boundary>Matching>Dependent** after the Independent boundary has been set up). When assigning an independent boundary, an **Independent Boundary** window appears. Otherwise a **Dependent Boundary** window appears.
4. Build up a tiny coordinate system of the face by setting the U vector and V vector to indicate the direction of the face. The U vector needs to be set using an arrow on the coordinate, and then the V vector is set automatically on the face (you can choose **Reverse Direction** of default V vector). Note that the U vectors and V vectors of a related independent and

dependent must have the same orientation.

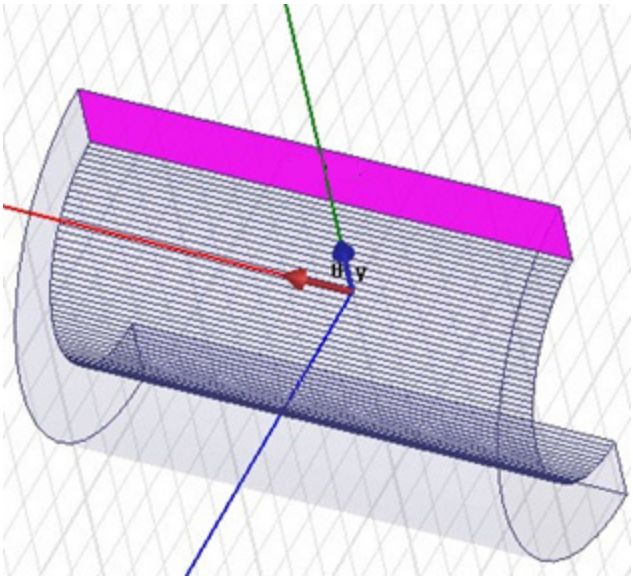


Figure 9-19 Set up U vector and V vector

5. When creating a dependent boundary, you also need to set its independent boundary and its magnetic field strength (either Hm or –Hm).
6. Click **OK** to finish independent or dependent boundary creation.

Creating Core Regions

Most core UDPs, which are covered in UDPs for Slot Cores and Coils, UDPs for Pole Cores and Coils, and UDPs for Other Cores and Coils, can be changed to core regions. The ways to change them into regions are similar. This section gives a general introduction of how to change those core UDPs into core regions.

Inner and outer cores are defined as follows:

- An inner core is a core for which the **DiaGap** value is greater than the **DiaYoke** value.
- An outer core is a core for which the **DiaGap** value is less than the **DiaYoke** value.

Creating an Inner Core Region

Because the properties and parameters of an inner core region should be the same as the corresponding inner core, an inner core region is usually created by replicating an inner core and changing the new object's **InfoCore** property to **100**. The detailed steps are as follows:

1. Select the inner core in the history tree. Then right-click to choose **Edit>Copy**.

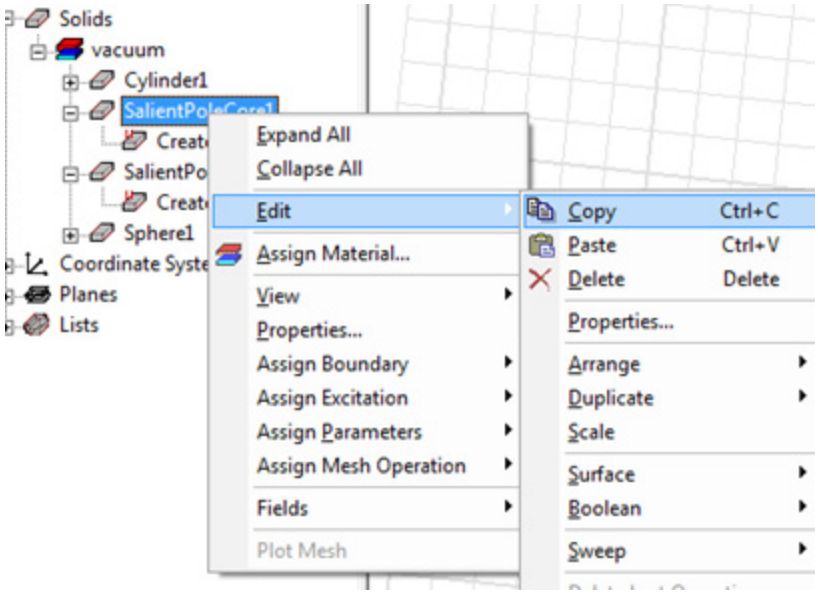


Figure 9-20 Copy a SalientPoleCore

2. Right-click the same object in the history tree to choose **Edit>Paste**. This process is shown in the following figure:

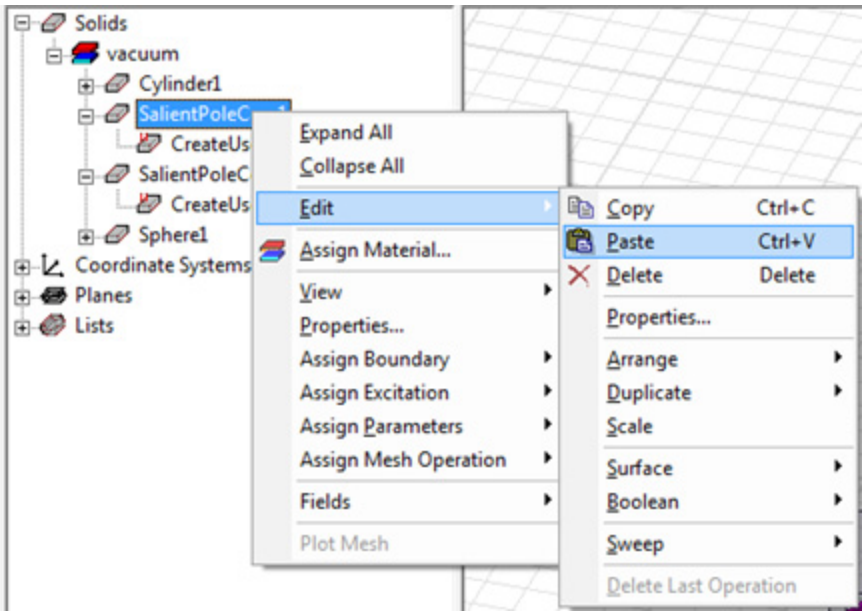


Figure 9-21 Paste a SalientPoleCore

3. Click **CreateUserDefinedPart**, and set **InfoCore** property to **100** in the pop-up window.

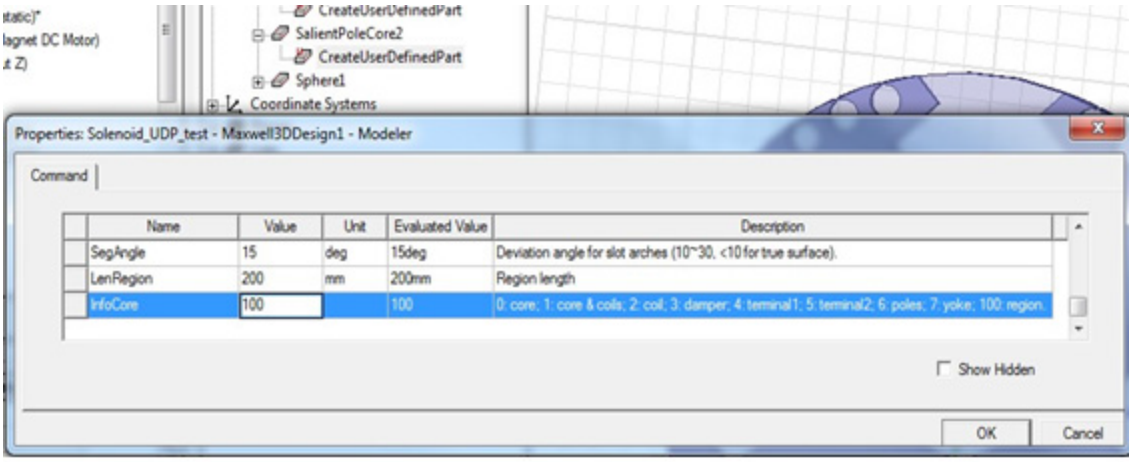


Figure 9-22 Change the value of InfoCore of SalientPoleCore

4. Click **OK.**

Creating an Outer Core Region

Because the properties and parameters of an outer core region should be the same as the corresponding outer core, an outer core region is usually created by replicating an outer core and changing the new object's **InfoCore** property to **100**. The detailed steps are as follows:

1. Select the outer core in the history tree. Then right-click to choose **Edit>Copy**.

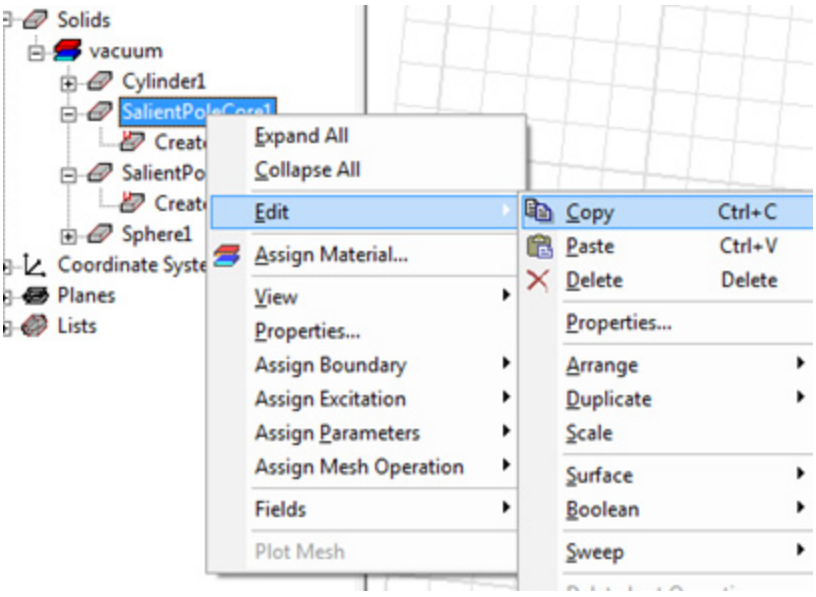


Figure 9-23 Copy a SalientPoleCore

2. Right-click the same object in the history tree to choose **Edit>Paste**. This process is shown in the following figure:

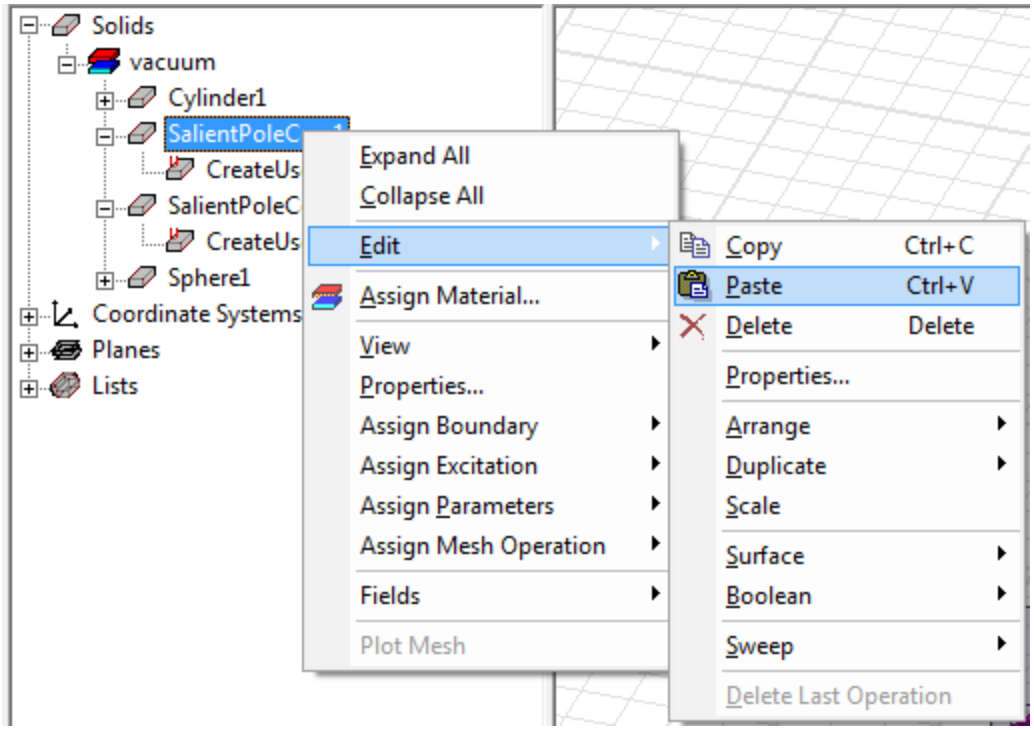


Figure 9-24 Paste a SalientPoleCore

3. Click **CreateUserDefinedPart**, and set **InfoCore** property to **100** in the pop-up window.

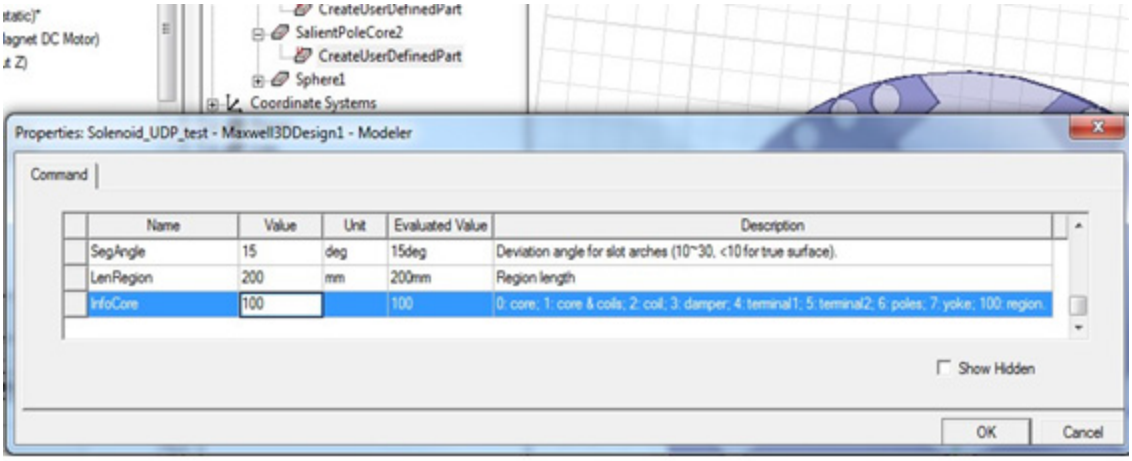


Figure 9-25 Change the value of InfoCore of SalientPoleCore

4. Click **OK.**

UDPs for Slot Cores and Coils

UDPs for defining slot cores and coils include:

- SlotCore UDP
- VentSlotCore UDP
- RacetrackSlotCore UDP
- NonSalientPoleCore UDP
- LapCoil UDP
- ConCoil UDP
- WaveCoil UDP
- SquirrelCage UDP
- DoubleCage UDP
- CupCoil UDP

SlotCore UDP

The SlotCore UDP is used to create a stator or rotor core with a distributed AC winding, or a DC commutating winding, or a squirrel-cage winding for the following machine types:

- Single-phase or multi-phase induction machines
- Wound-field or PM synchronous machines
- Brush-commutating machines, such as wound-field or PM DC machines, as well as universal motors
- Brushless DC machines, and others

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores.
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores.
Length	Core length.
Skew	Skew angle in core length range.
Slots	Number of slots.
SlotType	Slot Type: 1 to 6.
Hs0	Slot opening height.
Hs01	Slot closed bridge height.
Hs1	Wedge height.
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.

Property	Description
FilletType	0: a quarter circle; 1: a tangent connection; 2&3: arc bottom; 4&5: V bottom.
HalfSlot	0 for symmetric slot; 1 for half slot.
SegAngle	Deviation angle for slot arches (10~30, <10 for true surface).
LenRegion	Region length.
InfoCore	0: Core; 100: region.

These parameters are used in the following 2D SlotCore example:

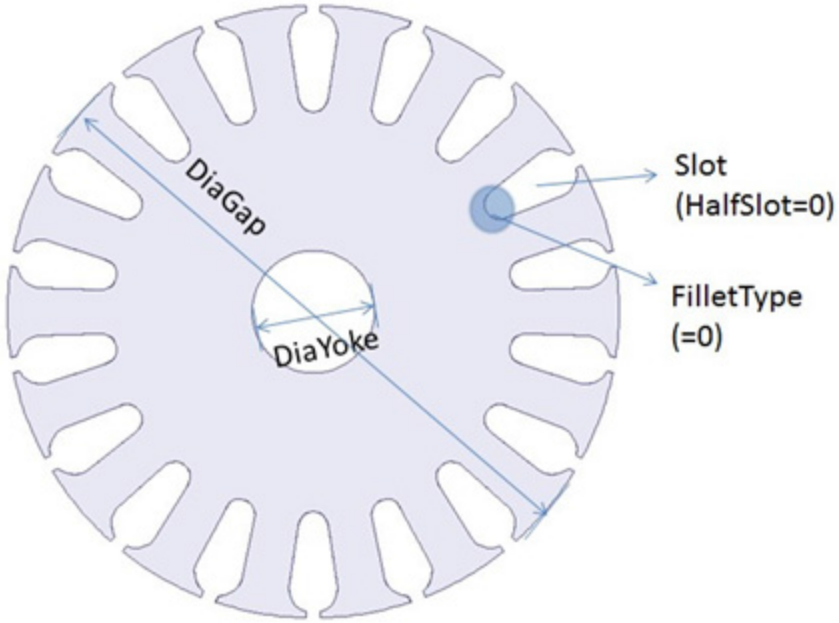


Figure 9-26 Diameters of SlotCore when HalfSlot is 0

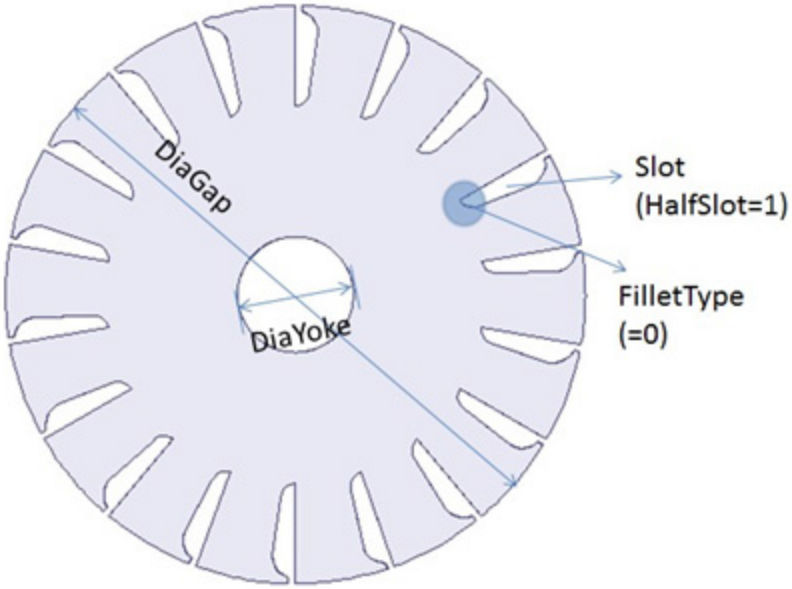


Figure 9-27 Diameters of SlotCore when HalfSlot is 1

3D figures of a slot core are shown in the following examples:

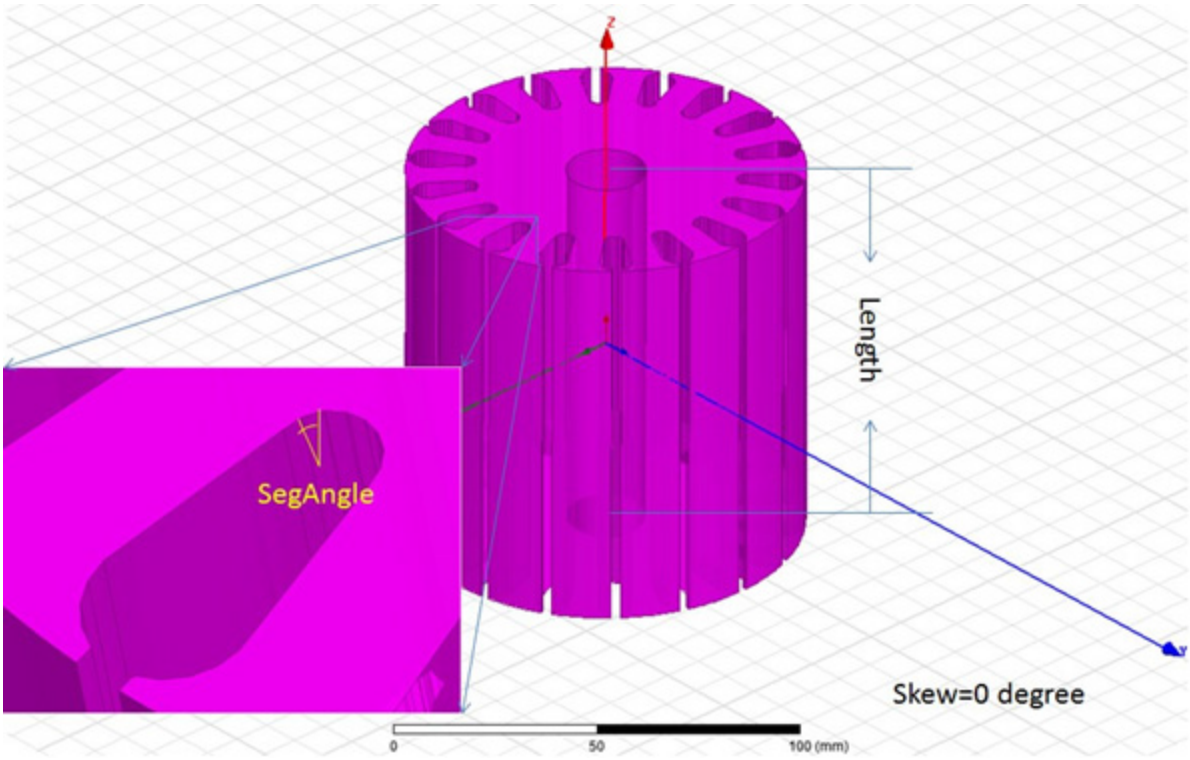


Figure 9-28 SegAngle and Length of SlotCore

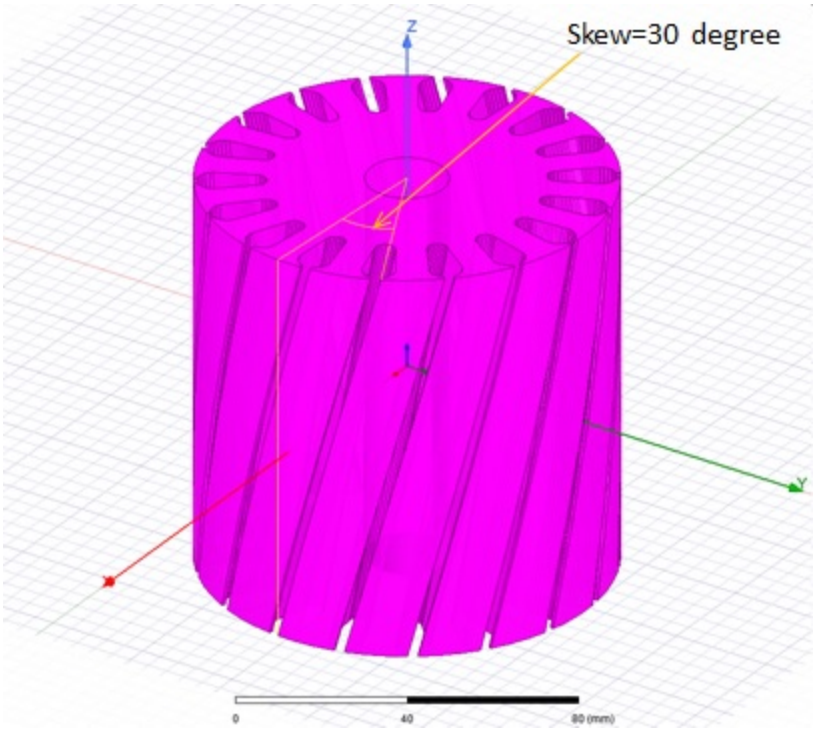


Figure 9-29 Skew of SlotCore

Six common slot types and their related parameters are shown in the following figures:

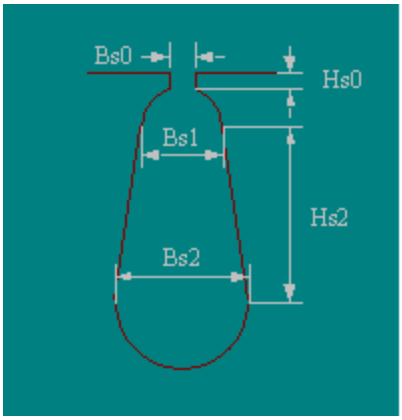


Figure 9-30 Slot Type 1

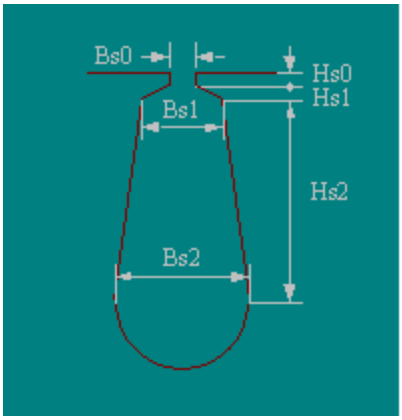


Figure 9-31 Slot Type 2

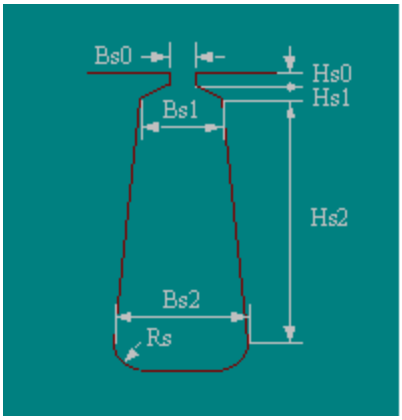


Figure 9-32 Slot Type 3

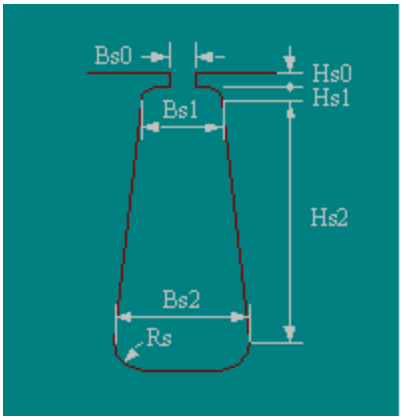


Figure 9-33 Slot Type 4

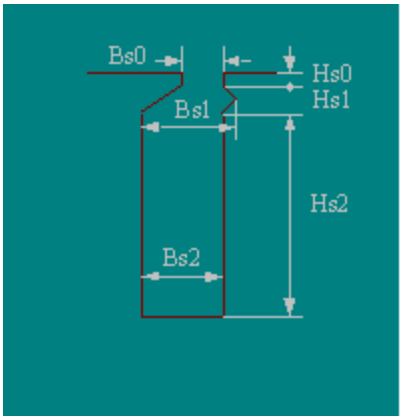


Figure 9-34 Slot Type 5

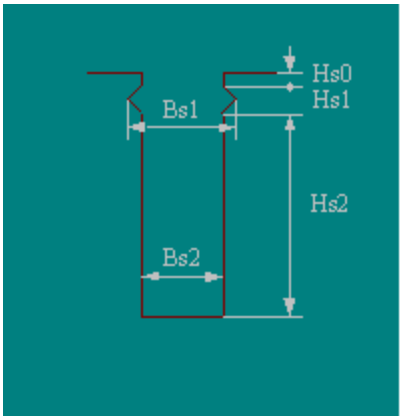


Figure 9-35 Slot Type 6

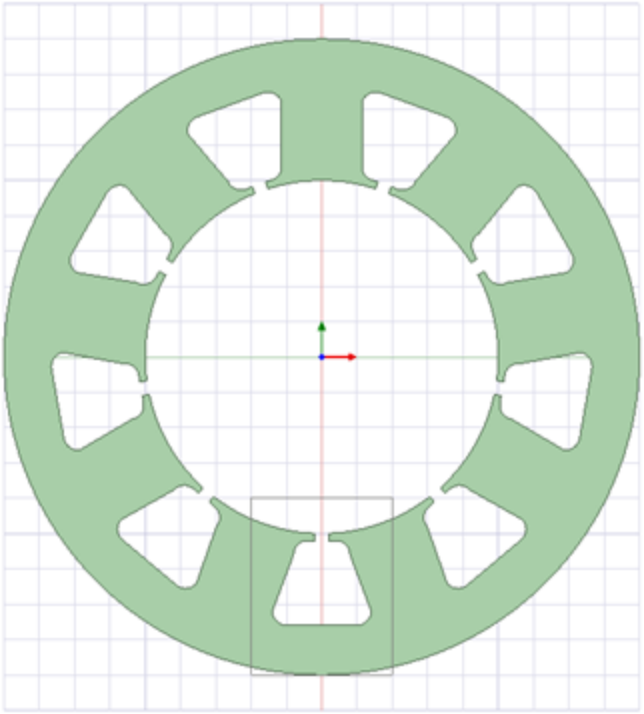
Slot cores or coils with a **SlotType** parameter can choose one of these slots.

Creating an Inner or Outer Slot Core

Set the value of **DiaYoke** and **DiaGap** as **DiaYoke**<**DiaGap** to create inner cores or **DiaGap**<**DiaYoke** for outer cores. You can set the values either when creating the core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

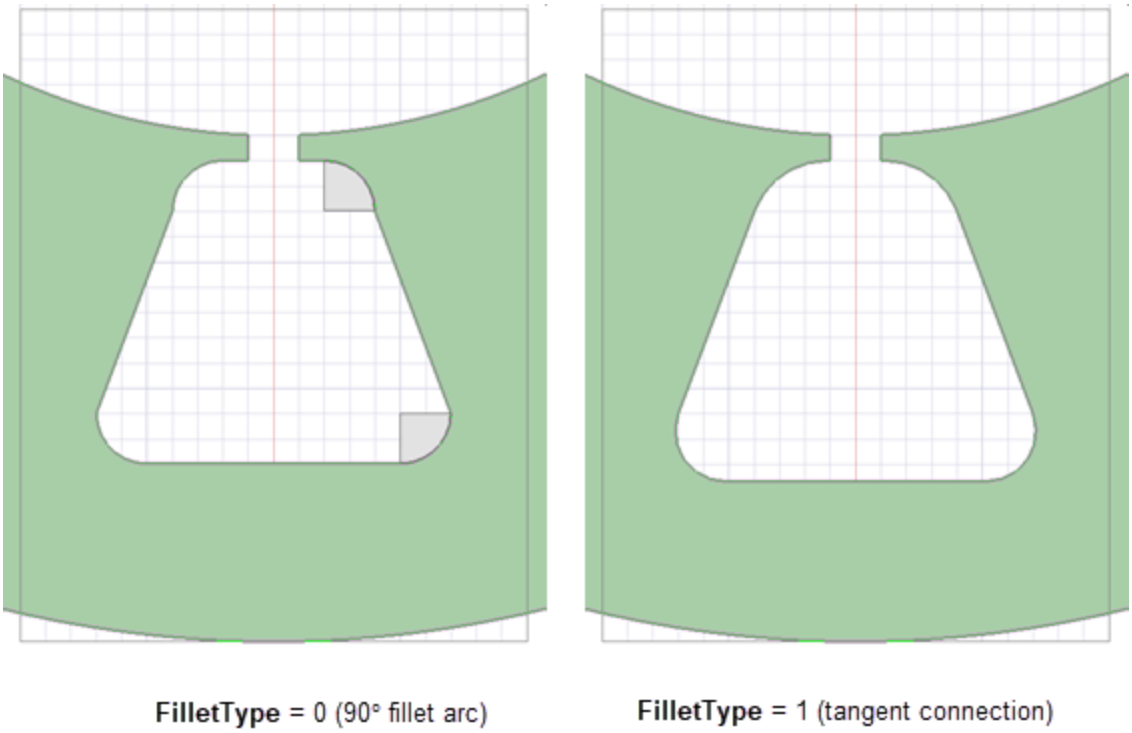
Set the **InfoCore** value to 0.

Slots with Different Fillet and Bottom Types



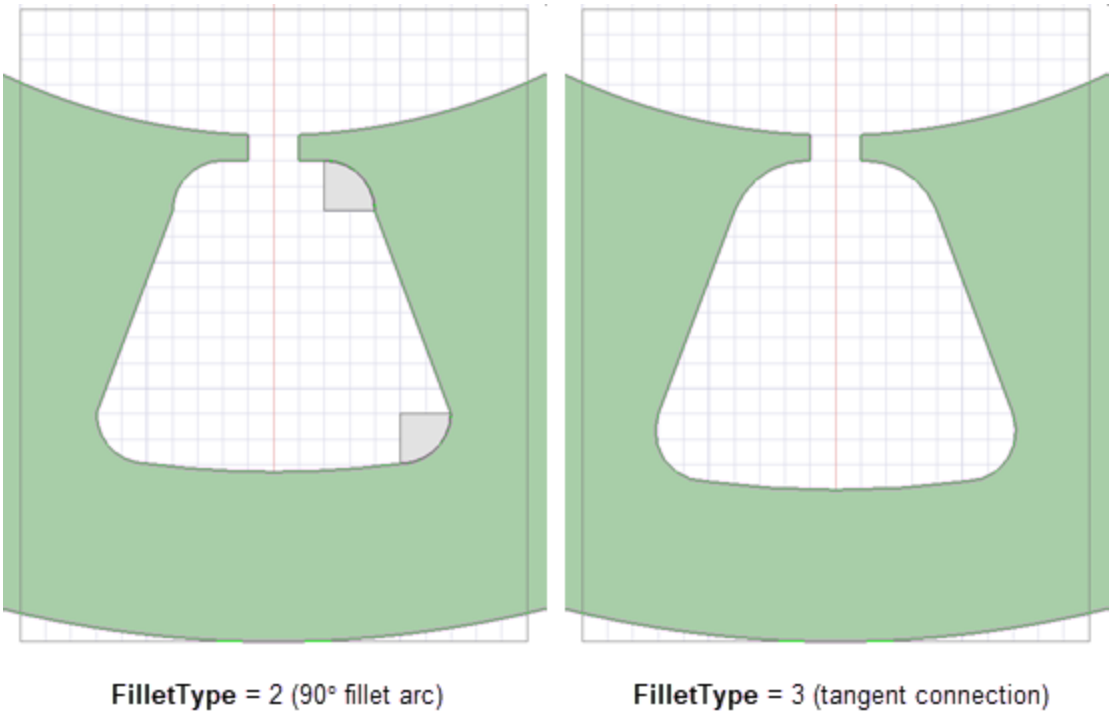
Slot core with Type-4 slots

Various values of **FilletType** are used for different fillet and bottom types. **FilletType** values 0 and 1 are for half-circle slot bottom for **SlotType** = 1 and 2, or for flat slot bottom for **SlotType** = 3 and 4, as shown below.



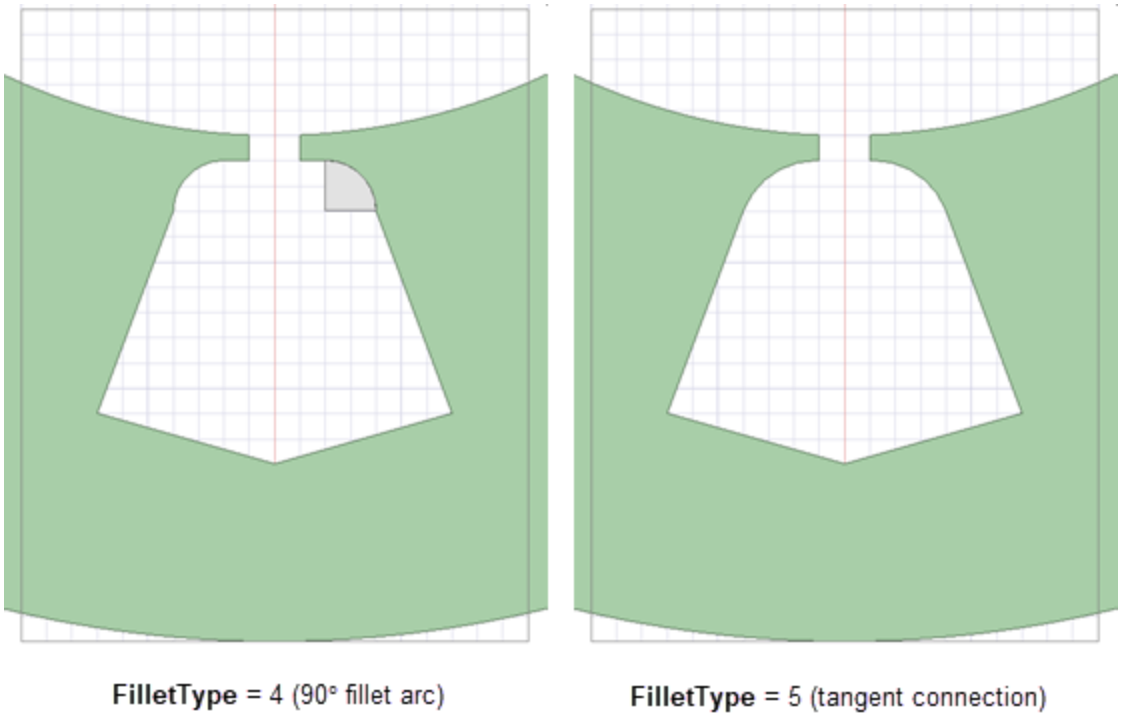
Slot Type 4 with flat bottom

FilletType values 2 and 3 are for outer-core arc slot bottom for **SlotType** = 3 and 4 only, as shown below.



Slot Type 4 with arc bottom

FilletType values 4 and 5 are for V-shape slot bottom for **SlotType** = 3 and 4 only, as shown below.



Slot Type 4 with V-shape bottom (Rs is V-shape height)

VentSlotCore UDP

The VentSlotCore UDP can create the same core that the SlotCore UDP does. However, the VentSlotCore UDP can create cores with radial vents and axial holes. If you want to create a slotted core without radial vents or axial holes, use the SlotCore UDP to have less data input.

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores.
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores.
Length	Core length.
Skew	Skew angle in core length range.
Slots	Number of slots.
SlotType	Slot Type: 1 to 6.
Hs0	Slot opening height.
Hs01	Slot closed bridge height.
Hs1	Wedge height.

Property	Description
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.
FilletType	0: a quarter circle; 1: a tangent connection; 2&3: arc bottom.
HalfSlot	0 for symmetric slot; 1 for half slot.
VentHoles	Number of axial vent holes per row.
HoleDiaIn	Diameter of inner vent holes.
HoleDiaOut	Diameter of outer vent holes.
HoleLocIn	Diameter of inner vent hole center layout circle.
HoleLocOut	Diameter of outer vent hole center layout circle.
VentDucts	Number of radial vent ducts.
DuctWidth	Axial width of radial vent ducts.
DuctPitch	Center-to-center distance between two adjacent ducts.
SegAngle	Deviation angle for slot arches (10~30, <10 for true surface).
LenRegion	Region length.
InfoCore	0: Core; 100: region.

These parameters are used in the following inner VentSlotCore example:

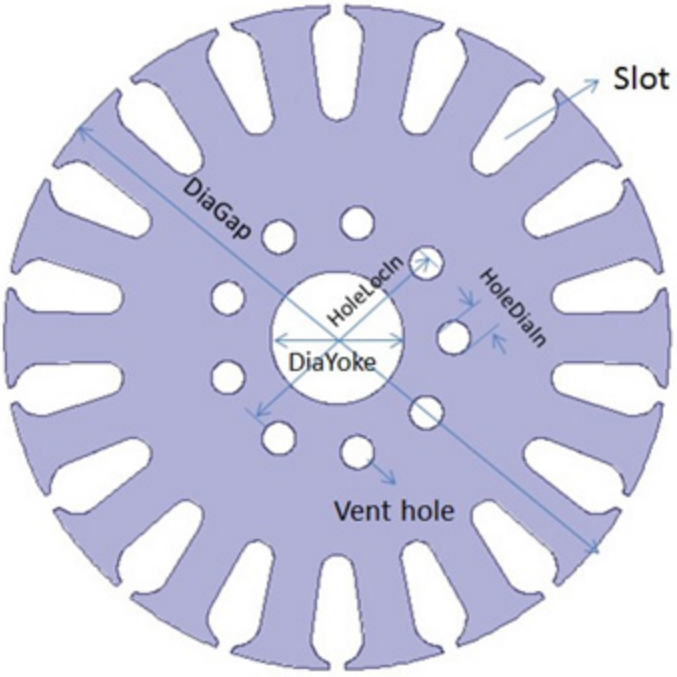


Figure 9-36 Parameters on cross-section of VentSlotCore

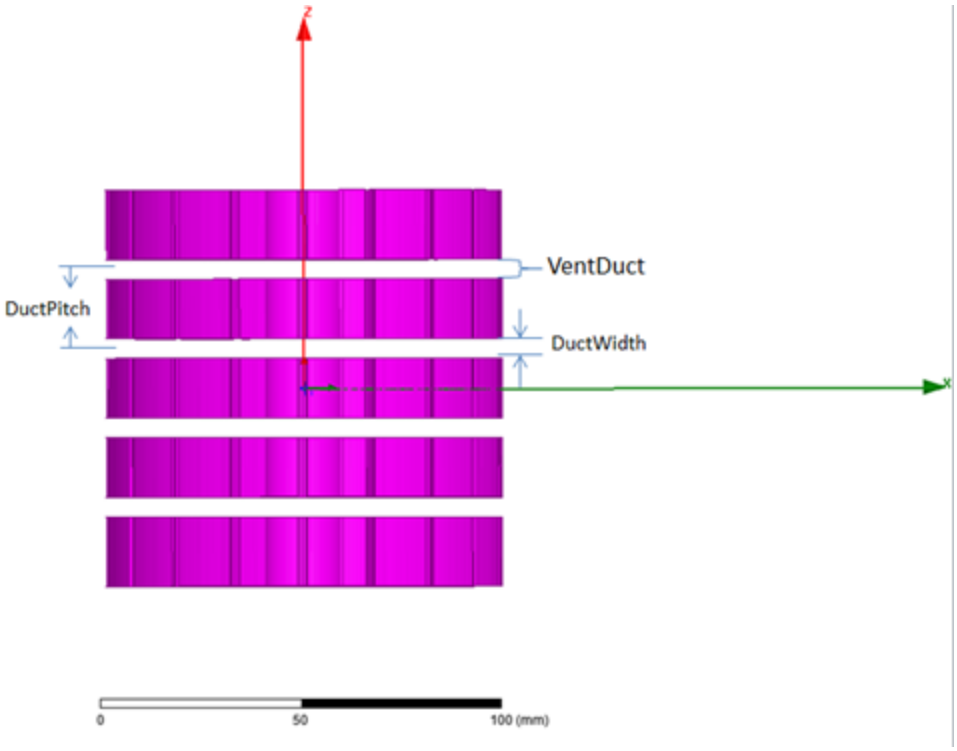


Figure 9-37 Parameters on axial direction of VentSlotCore

Note	For parameters such as **Skew**, **FilletType**, **SegAngle**, **HalfSlot** and parameters related to slots, refer to SlotCore UDP parameters.

Creating an Inner or Outer Vent Slot Core

Set the value of **DiaYoke** and **DiaGap** as **DiaYoke**<**DiaGap** to create inner cores or **DiaGap**<**DiaYoke** for outer cores. You can set the values when creating the core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Set the **InfoCore** value to 0.

RacetrackSlotCore UDP

The RacetrackSlotCore UDP is used to create a stator core with a single-phase concentric AC winding for single-phase induction machines. The stator may have up to three different slot sizes.

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores.
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores.
Length	Core length.
Skew	Skew angle in core length range.
Slots	Number of slots.
SlotType	Slot Type: 1 to 6.
Hs0	Slot opening height.
Hs01	Slot closed bridge height.
Hs1	Wedge height.
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.
FilletType	0: a quarter circle; 1: a tangent connection; 2&3: arc bottom.
Poles	Number of poles.
WidYoke	Racetrack width for 2/4-pole cores.
MidSlots	Number of middle-size slots for 2/4-pole cores.

Property	Description
MidHs2	Middle-size slot body height for 2/4-pole cores.
MidBs2	Middle-size slot body bottom width for 2/4-pole cores.
SmlSlots	Number of small-size slots for 2/4-pole cores.
SmlHs2	Small-size slot body height for 2/4 pole cores.
SmlBs2	Small-size slot body bottom width for 2/4-pole cores.
SegAngle	Deviation angle for slot arches (10~30, <10 for true surface).
LenRegion	Region length.
InfoCore	0: Core; 100: region.

These parameters are used in the following outer Racetrack SlotCore example:

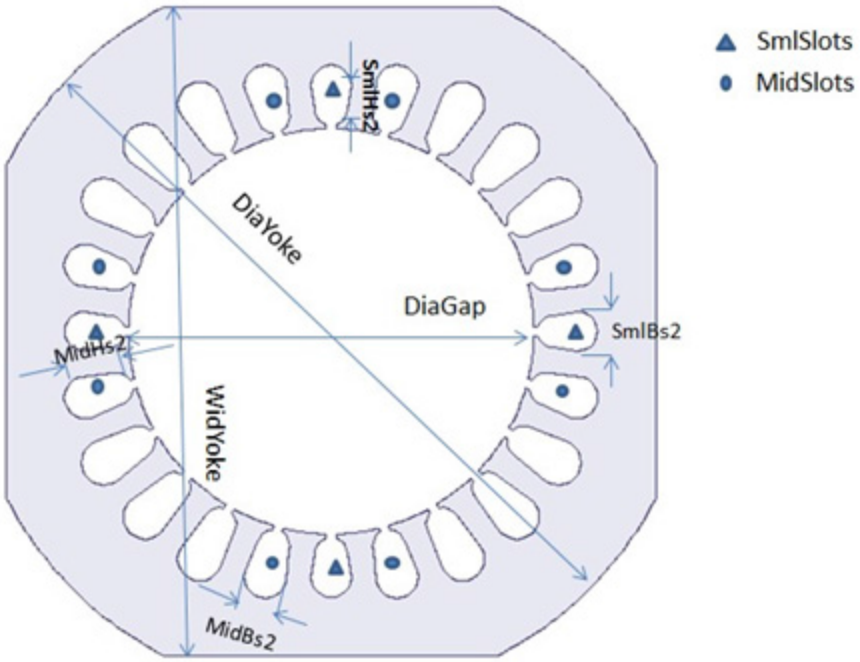


Figure 9-38 Parameters on cross-section of RacetrackSlotCore

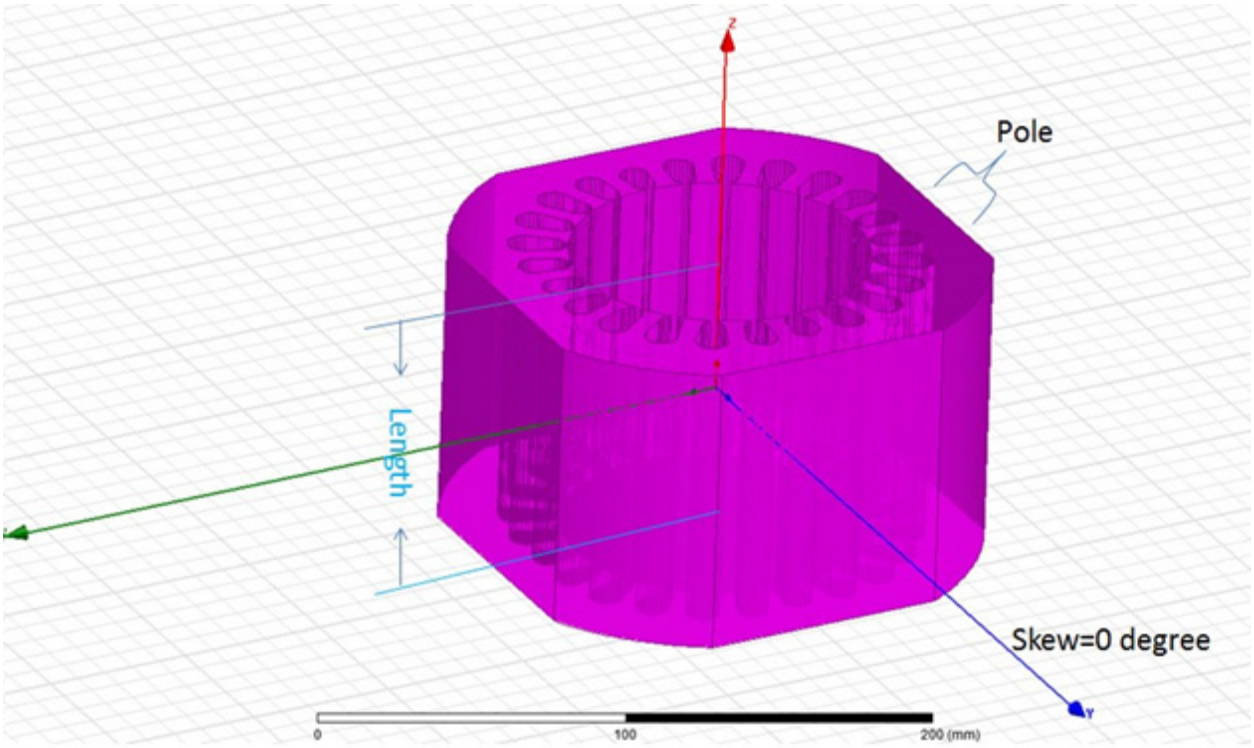


Figure 9-39 Parameters on axial direction of RacetrackSlotCore

Note	For parameters such as **Skew**, **FilletType**, **SegAngle** and parameters related to slots, refer to SlotCore UDP parameters.

Creating an Outer Racetrack Slot Core

Generally, Racetrack Slot Cores are used as outer cores. Set the value of **DiaYoke** and **DiaGap** as **DiaGap**<**DiaYoke** for outer cores. You can set the values when creating the core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Set the **InfoCore** value to 0.

NonSalientPoleCore UDP

The NonSalientPoleCore UDP is used to create non-salient rotor cores for non-salient rotor synchronous machines. This core type may include surface tangential vent ducts, axial vent ducts, as well as a damper.

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores.
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner

	cores.
Length	Core length.
Skew	Skew angle in core length range.
IndexingSlots	Number of indexing slots for slot pitch calculation.
ActualSlots	Number of actual slots: ActualSlots <= IndexingSlots.
SlotType	Slot Type: 1 to 7.
Hs0	Slot opening height.
Hs1	Slot wedge height.
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.
FilletType	0: a quarter circle; 1: a tangent connection; 2&3: arc bottom.
Poles	Number of poles.
TangentDucts	Number of surface tangential vent ducts.
TDuctWidth	Axial width of surface tangential vent ducts (or damper rings).
TDuctDepth	Radial depth of surface tangential vent ducts (or damper rings).
TDuctPitch	Pitch of surface tangential vent ducts (or damper bar extension).
AxialDucts	Number of axial vent ducts per pole.
ADuctWidth	Width of axial vent ducts in main tooth.
ADuctDepth	Depth of axial vent ducts in main tooth.
SegAngle	Deviation angle for slot arches (10~30, <10 for true surface).
LenRegion	Region length.
InfoCore	0: Core; 100: region.

These parameters are used in the following inner Non-Salient SlotCore example:

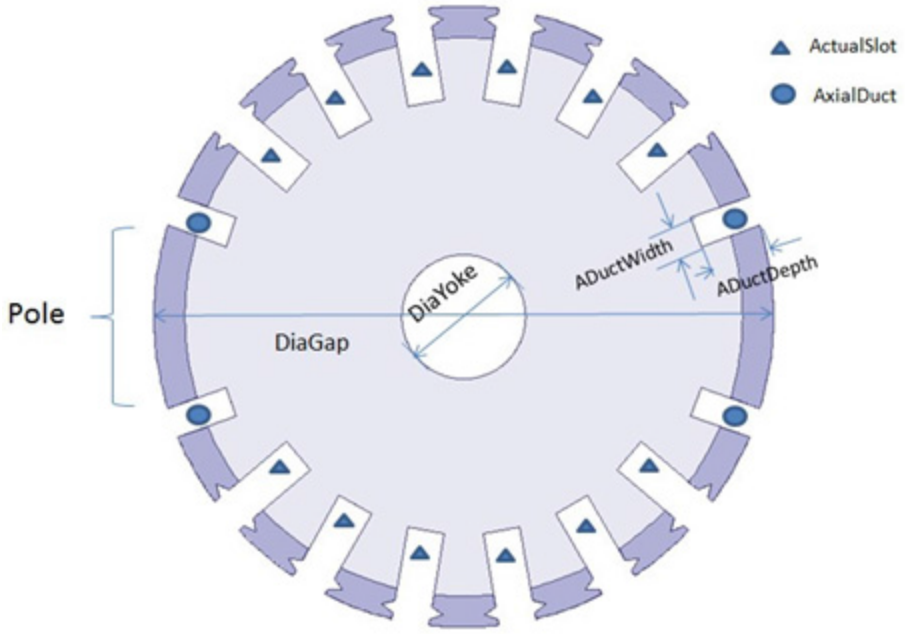


Figure 9-40 Parameters on cross-section of NonSalientPoleCore

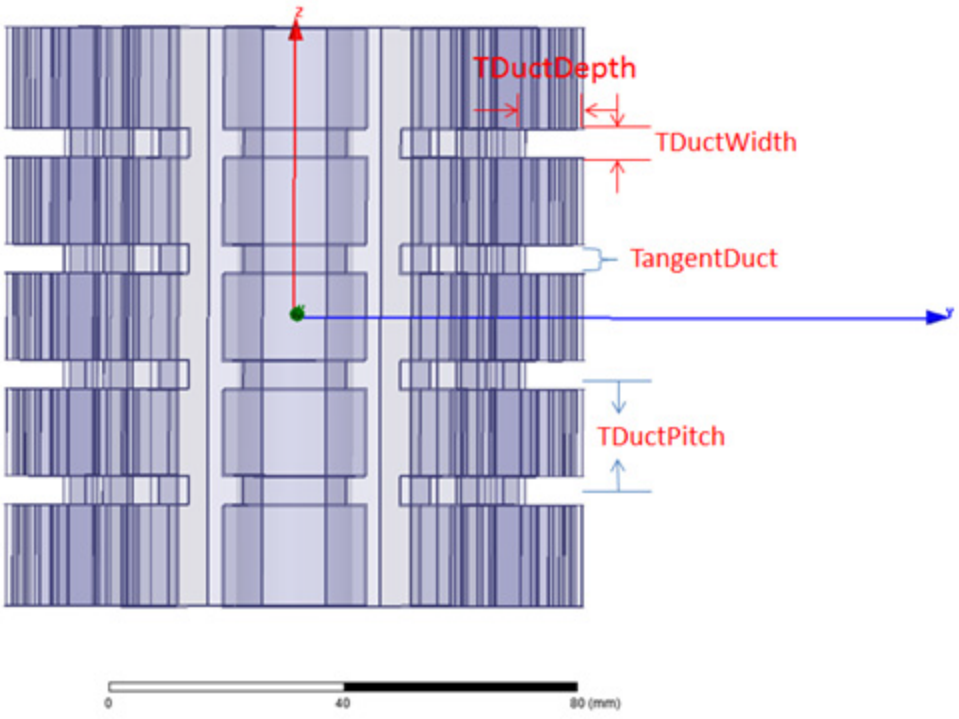


Figure 9-41 Parameters on axial direction of NonSalientPoleCore

Note	For parameters such as **Skew**, **FilletType**, **SegAngle** and parameters related to slots, refer to SlotCore UDP parameters.

Creating an Inner Non-Salient Pole Core

Generally, Non-Salient Pole Cores are used as inner cores. Set the value of **DiaYoke** and **DiaGap** as **DiaGap**>**DiaYoke** for inner cores. You can set the values when creating the core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Set the **InfoCore** value to 0.

Creating a Non-Salient Pole Core with a Q-Axis Damper

In general, this process is realized by a subtraction of two objects. Several steps are needed to finish the procedure.

1. Click **Project>Insert Maxwell 2D Design** to create a Maxwell2D Design in a project.

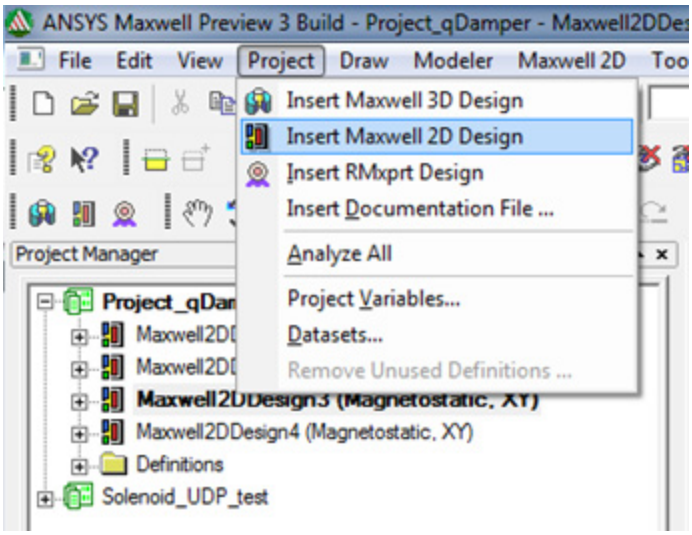


Figure 9-42 Insert a 2D design to create a Non-Salient Pole Core

2. Click **Draw>User Defined Primitive>RMxprt>NonSalientPoleCore** to draw a Non-Salient Pole Core UDP. In order to show it in a 2D dimension, set the **Length** to 0. Also set

AxialDucts to 0 for subtraction, as shown in the following figure.

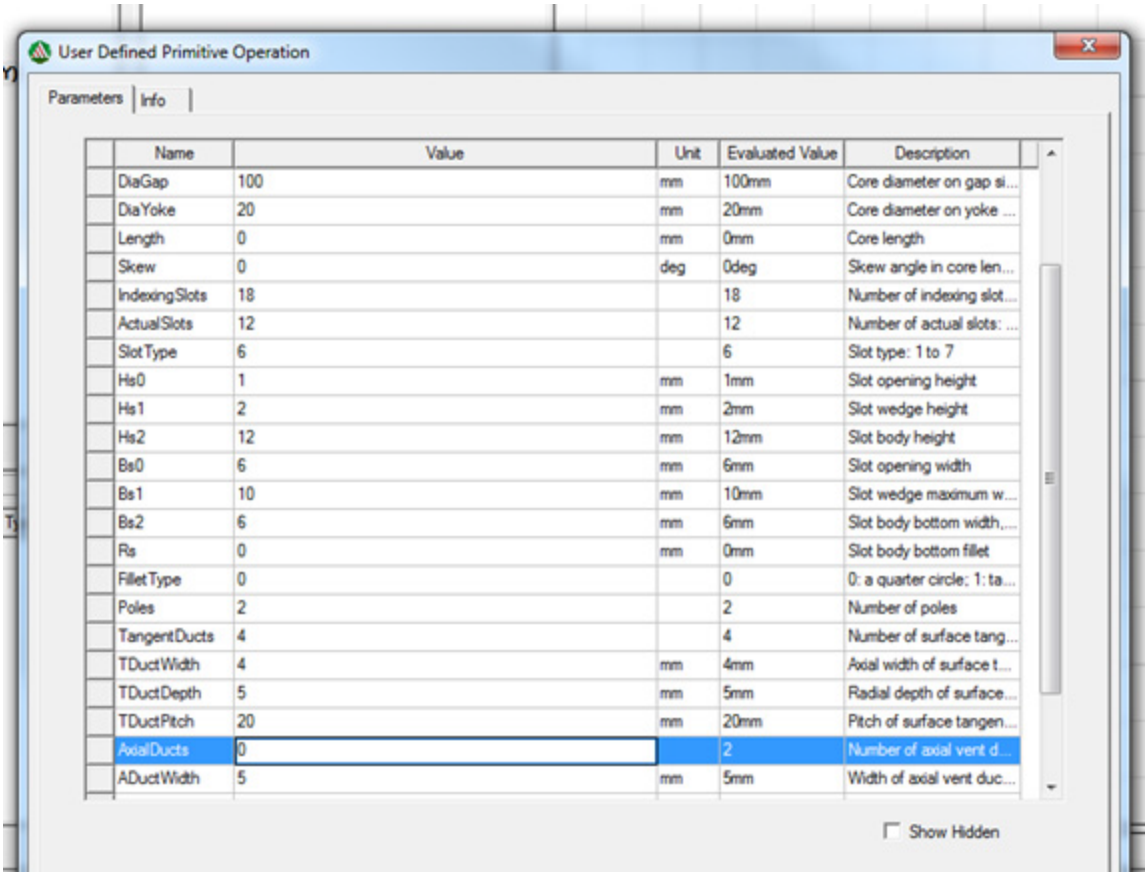
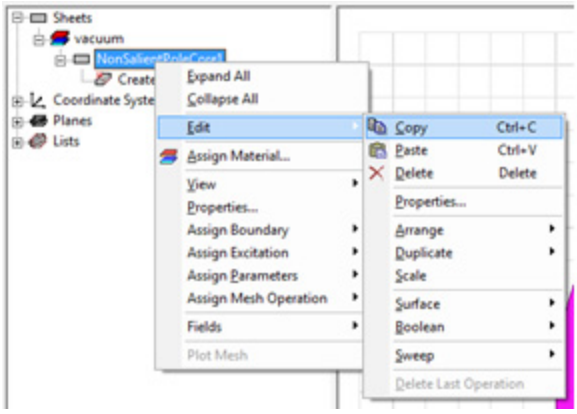


Figure 9-43 Parameter settings of NonSalientPoleCore

3. Create a damper. Make a copy of the NonSalientPoleCore, and then paste it and rename it to damper.



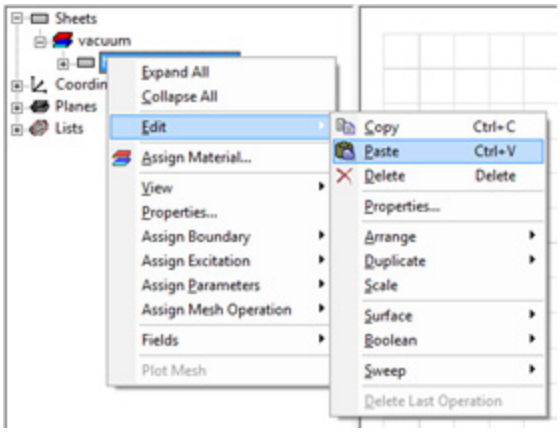


Figure 9-44 Make a copy of NonSalientPoleCore and paste it

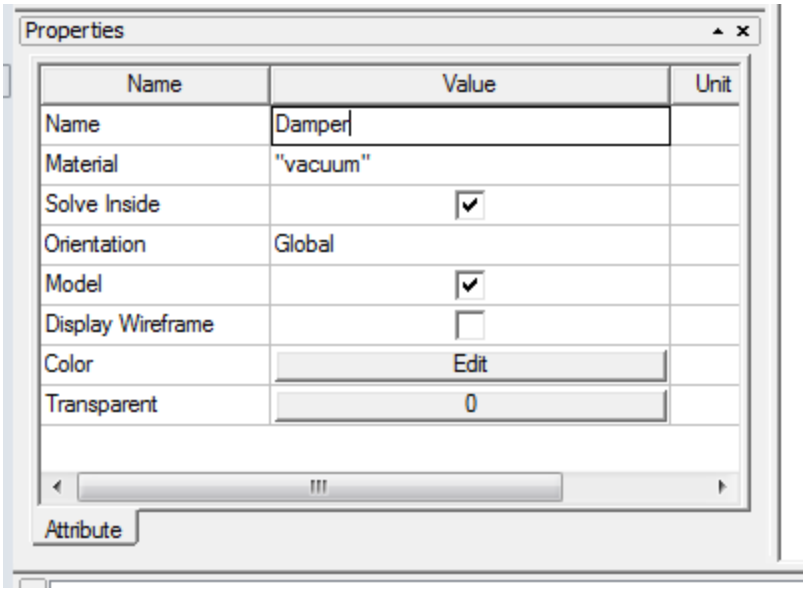
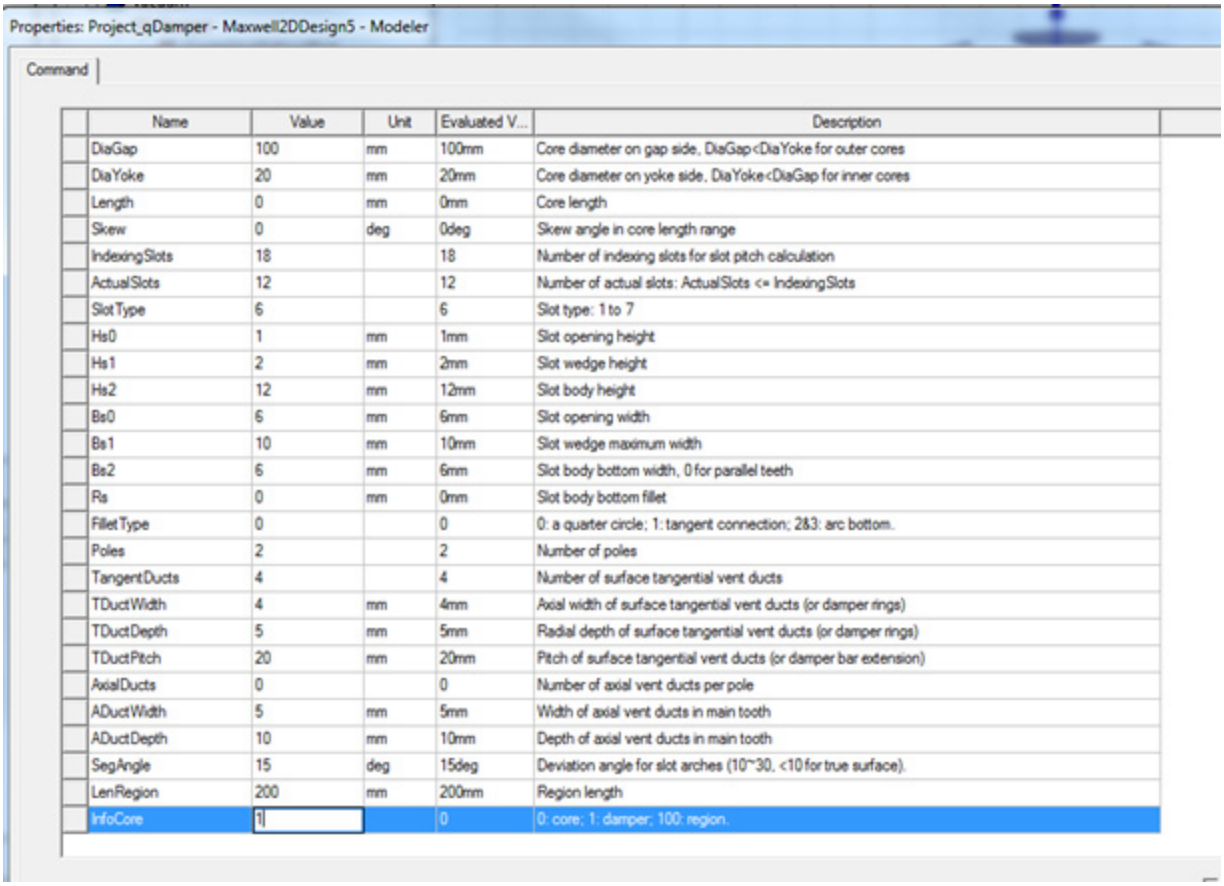


Figure 9-45 Rename the new NonSalientPoleCore to “Damper”

4. Change the **InfoCore** value to **1** to make a damper. Also change the **Color** of the damper to distinguish it from the NonSalientPoleCore.



Name	Value	Unit	Evaluated V...	Description
DiaGap	100	mm	100mm	Core diameter on gap side, DiaGap<DiaYoke for outer cores
DiaYoke	20	mm	20mm	Core diameter on yoke side, DiaYoke<DiaGap for inner cores
Length	0	mm	0mm	Core length
Skew	0	deg	0deg	Skew angle in core length range
IndexingSlots	18		18	Number of indexing slots for slot pitch calculation
ActualSlots	12		12	Number of actual slots: ActualSlots <= IndexingSlots
SlotType	6		6	Slot type: 1 to 7
Hs0	1	mm	1mm	Slot opening height
Hs1	2	mm	2mm	Slot wedge height
Hs2	12	mm	12mm	Slot body height
Bs0	6	mm	6mm	Slot opening width
Bs1	10	mm	10mm	Slot wedge maximum width
Bs2	6	mm	6mm	Slot body bottom width, 0 for parallel teeth
Rs	0	mm	0mm	Slot body bottom fillet
FilletType	0		0	0: a quarter circle; 1: tangent connection; 2&3: arc bottom.
Poles	2		2	Number of poles
TangentDucts	4		4	Number of surface tangential vent ducts
TDuctWidth	4	mm	4mm	Axial width of surface tangential vent ducts (or damper rings)
TDuctDepth	5	mm	5mm	Radial depth of surface tangential vent ducts (or damper rings)
TDuctPitch	20	mm	20mm	Pitch of surface tangential vent ducts (or damper bar extension)
AxialDucts	0		0	Number of axial vent ducts per pole
ADuctWidth	5	mm	5mm	Width of axial vent ducts in main tooth
ADuctDepth	10	mm	10mm	Depth of axial vent ducts in main tooth
SegAngle	15	deg	15deg	Deviation angle for slot arches (10~30, <10 for true surface).
LenRegion	200	mm	200mm	Region length
InfoCore	1		0	0: core; 1: damper; 100: region.

Figure 9-46 Change the InfoCore value to make it a damper

The objects appear as follows in the coordinate system window.

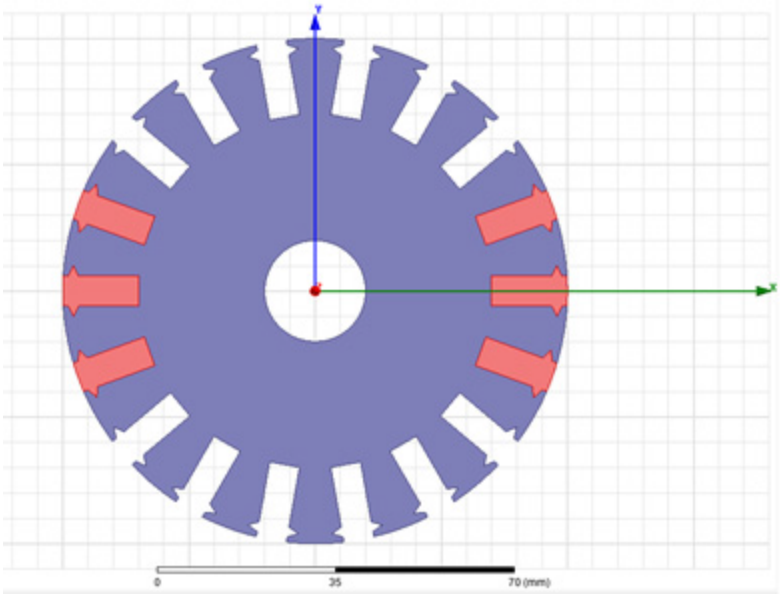


Figure 9-47 Non-Salient Pole Core (gray) with a damper (red)

5. You can also change the damper's **SlotType** and slot parameters if necessary. For instance, **SlotType** is reset to **1** and slot parameters are reassigned as in the following

figure.

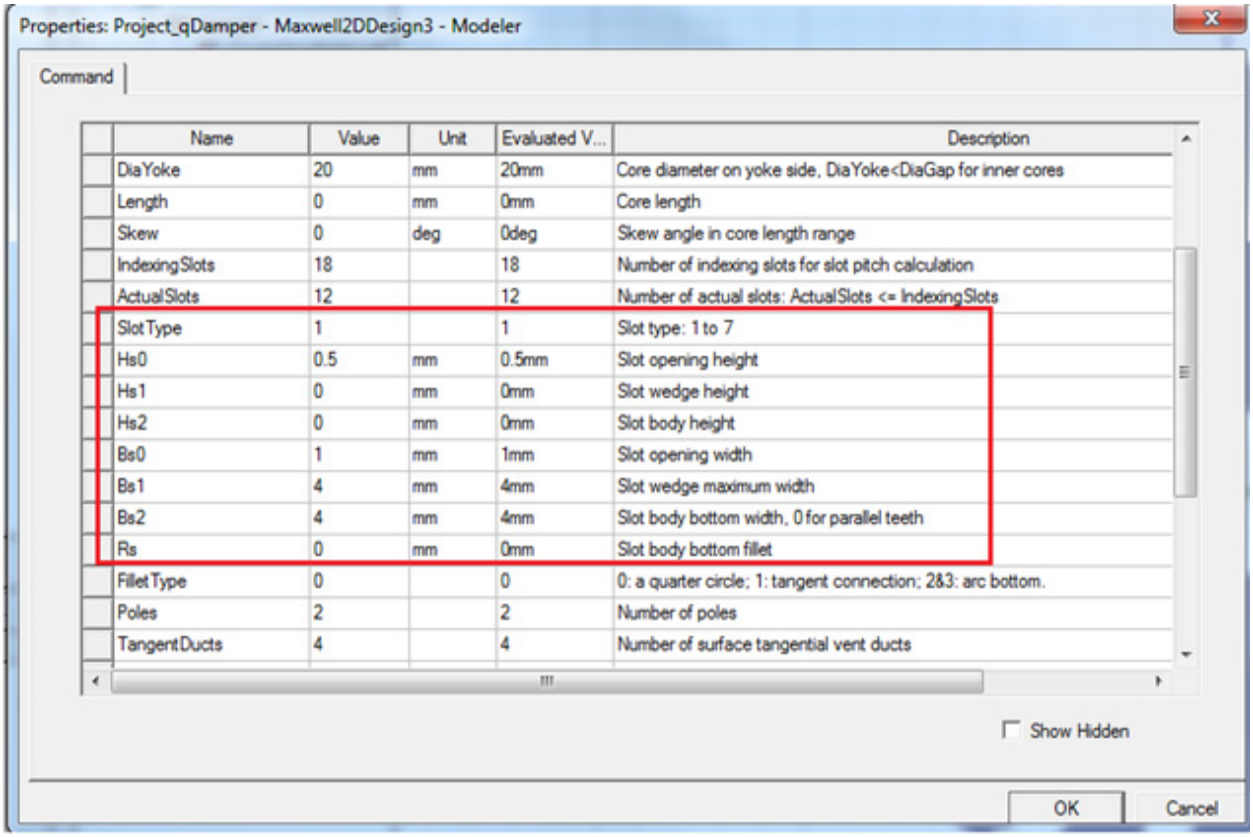


Figure 9-48 Change the slot-related parameters of the damper

After you modify the parameters, the objects in the coordinate system window appear as in the following example:

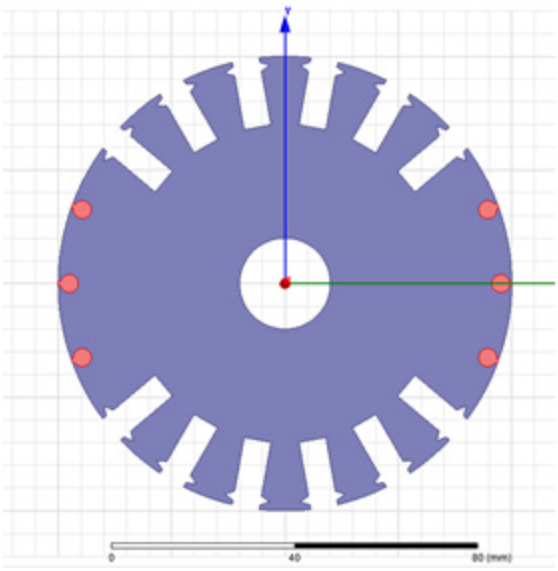


Figure 9-49 Results after changing the slot-related parameters of the damper

6. Select both objects (Damper and NonSalientPoleCore) in the history tree simultaneously by pressing **Ctrl** on the keyboard. Click **Modeler>Boolen>Subtract**. A Subtract window

appears.

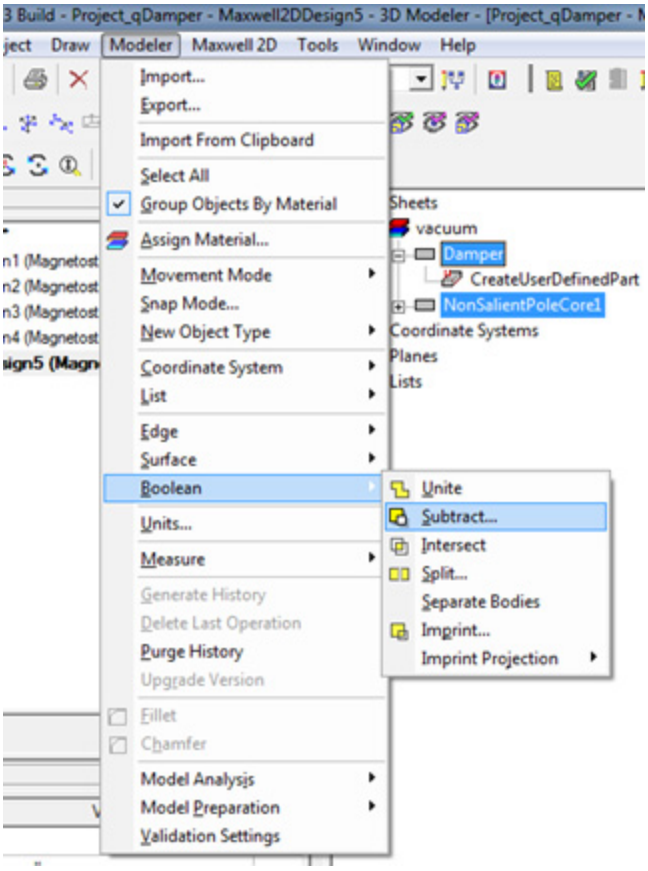


Figure 9-50 Subtraction of two objects

7. Move NonSalientPoleCore to **Blank Parts** and Damper to **Tool Parts**,and check the **Clone tool objects before operation** box. Click **OK** to finish subtraction.

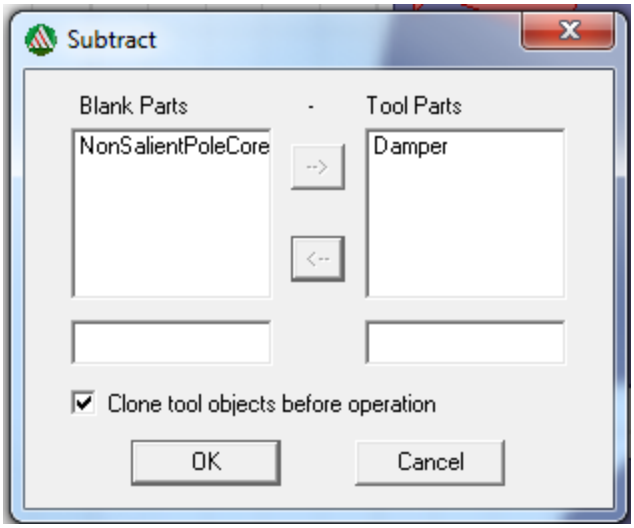


Figure 9-51 Set Blank Parts and Tool Parts

8. The final history tree and coordinate system window are shown in the following example:

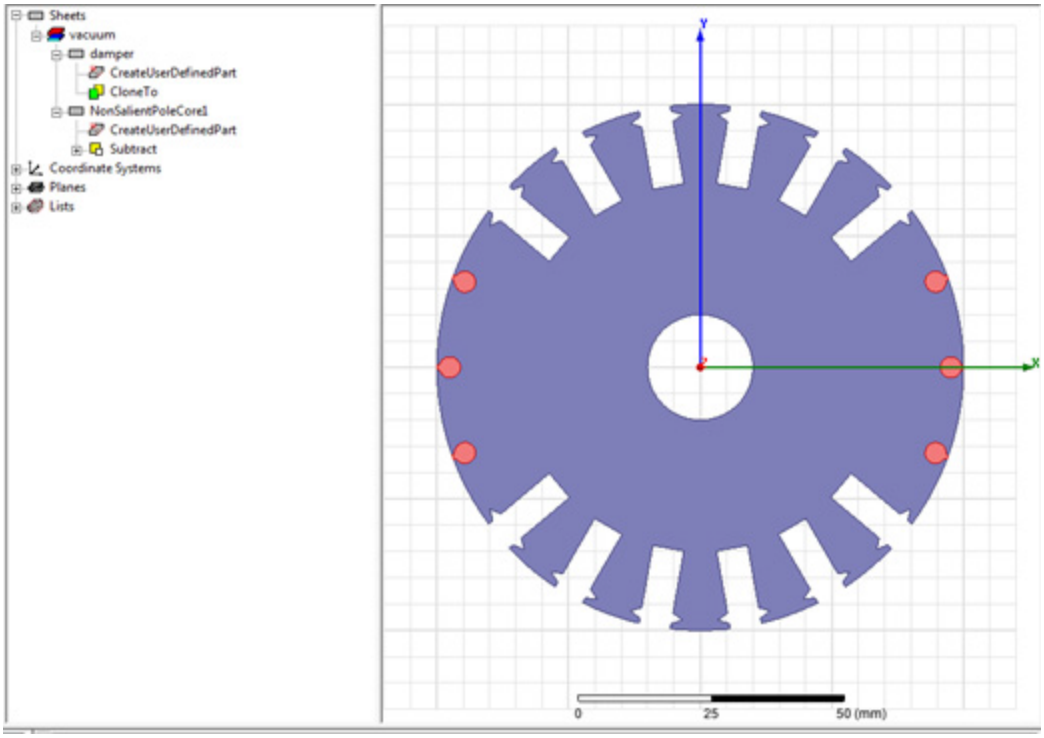


Figure 9-52 Objects after subtraction

Creating a Non-Salient Pole Core with a Uniform-Distributed Damper

This process relies on subtraction of two objects and is very similar to Creating a Non-Salient Pole Core with a Q-Axis Damper. The major difference is that you create a squirrel cage instead of copying a NonSalientPoleCore to create a damper.

1. Click **Project>Insert Maxwell 2D Design** to create a Maxwell2D Design in a project.
2. Click **Draw>User Defined Primitive>RMxprt>NonSalientPoleCore** to draw a Non-Salient Pole Core UDP. To show it in a 2D dimension, set the **Length** to 0. Also set **AxialDucts** to 0 for subtraction, as shown in Figure 9-53 .
3. Click **Draw>User Defined Primitive>RMxprt>SquirrelCage** to draw a squirrel cage UDP. Set the **Length** to 0, change its **Color** to a different one, and rename it to “Damper.” Then objects in the coordinate system window appear as in the following figure:

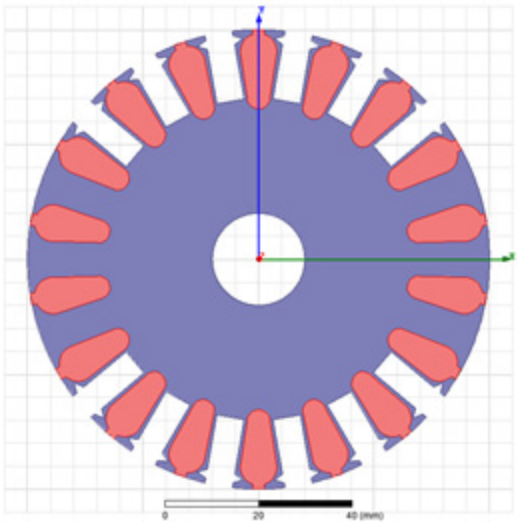


Figure 9-53 Non-Salient Pole Core with a damper (red)

4. You can also change the damper’s **SlotType** and slot parameters if necessary. For instance, **SlotType** is reset to **1** and slot parameters are reassigned as in the following

figure.

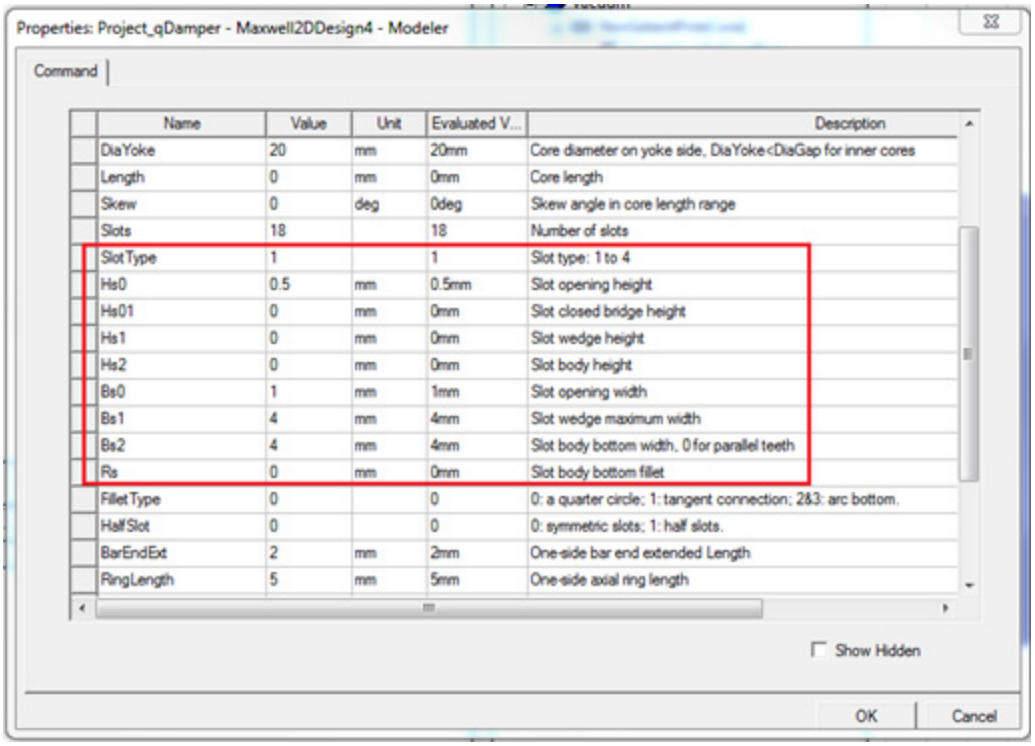


Figure 9-54 Change the slot-related parameters of the damper

5. After you modify the parameters, the objects in the coordinate system window appear as in the following figure:

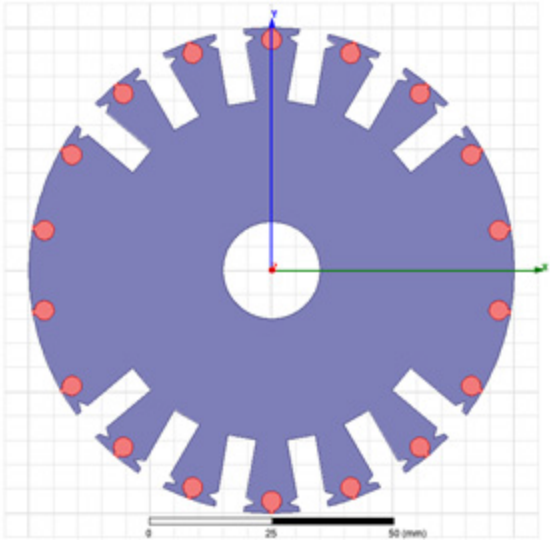


Figure 9-55 Results after changing the slot-related parameters of the damper

6. Select both objects (Damper and NonSalientPoleCore) in the history tree simultaneously by pressing **Ctrl** on the keyboard. Click **Modeler>Boolean>Subtract**. A Subtract window appears.
7. Move NonSalientPoleCore to **Blank Parts** and Damper to **Tool Parts**, and check the **Clone tool objects before operation** box. Click **OK** to finish subtraction.
8. The final history tree and coordinate system window are shown in the following figure:

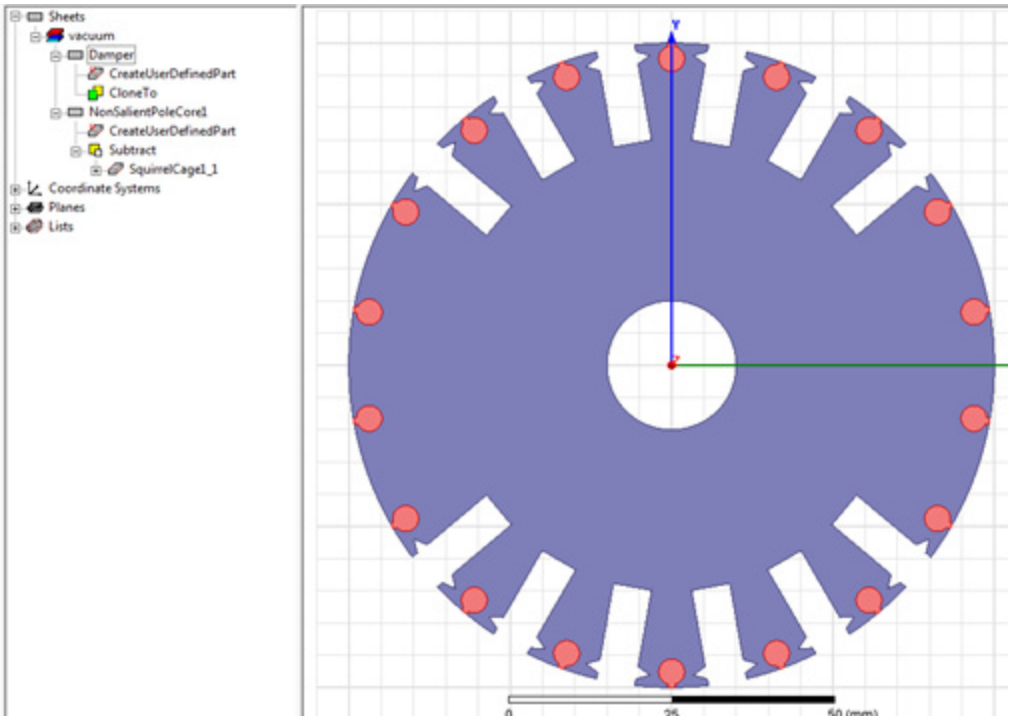


Figure 9-56 Non-Salient Pole Core with a uniform-distributed damper

LapCoil UDP

The LapCoil UDP is used to create a coil for lap-type distributed AC windings or lap-type DC commutating windings in a slotted core created by the SlotCore or VentSlotCore UDP. This UDP can also create a coil cross-section used as a terminal for current assignment.

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores.
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores.
Length	Core length.
Skew	Skew angle in core length range.
Slots	Number of slots.

Property	Description
SlotType	Slot Type: 1 to 7.
Hs0	Slot opening height.
Hs1	Slot wedge height.
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.
FilletType	0: a quarter circle; 1: a tangent connection; 2&3: arc bottom; 4&5: V bottom.
Layers	Number of winding layers.
CoilPitch	Coil pitch measured in slots.
EndExt	One-side end extended length.
SpanExt	Axial length of end span; 0 for no span.
BendAngle	Bending angle viewed in the rz plane.
SegAngle	Deviation angle for slot arches (5~15, <5 for true surface).
LenRegion	Region length.
InfoCoil	0: winding; 1: coil; 2 terminal1; 3: terminal2; 4:insulation; 100: region.

A single coil is much easier to be described with a slot core as reference. In the following two figures, coils are in gray and slot cores are orange. The lap coil's **InfoCoil** property is set to **1** to use a single coil to show parameters of lap coil.

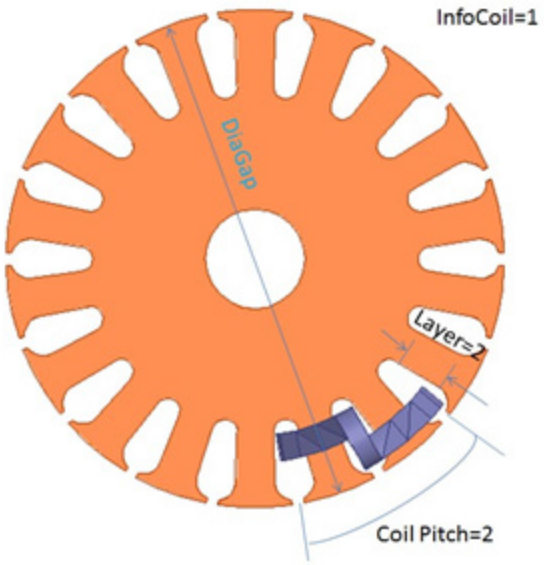


Figure 9-57 Single Lap Coil with Coil Pitch set to 2

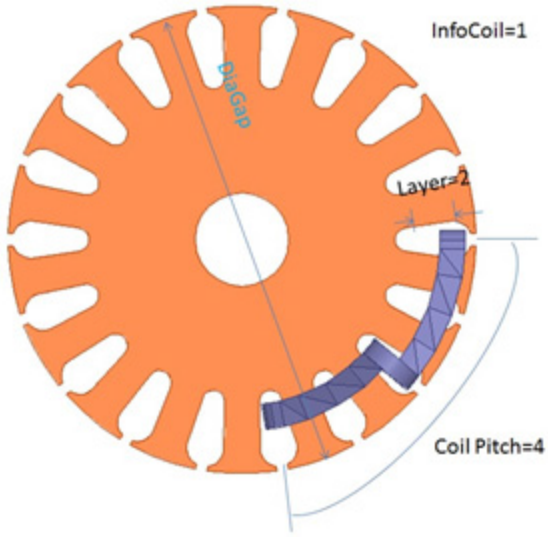


Figure 9-58 Single Lap Coil with Coil Pitch set to 4

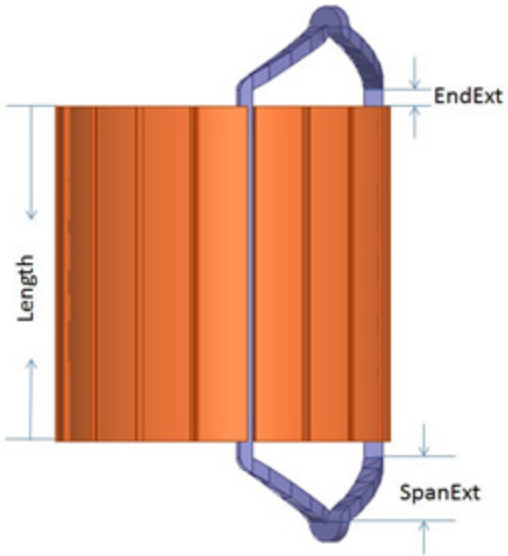


Figure 9-59 Axial Single Lap Coil with Coil Pitch set to 4

Figure 9-60

Figure 9-61 Below is an axial view of an outer core winding with bended end coil. The lap coil's InfoCoil property is set to 0.

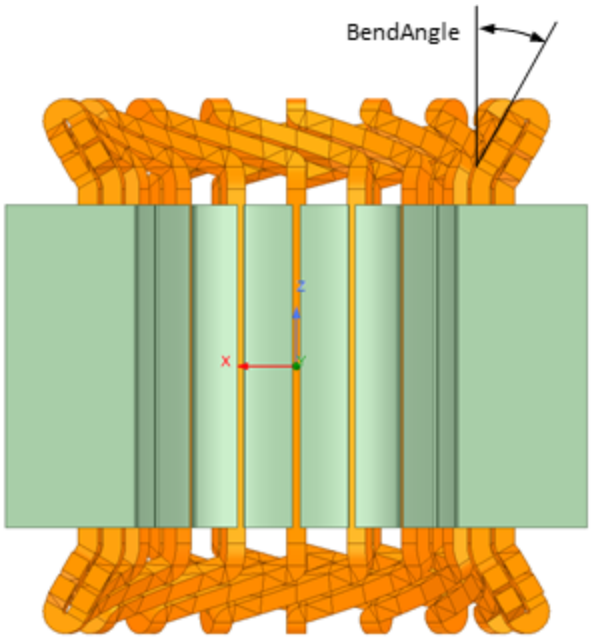


Figure 9-62

Creating a Whole Lap Winding

A whole lap winding is created by default or by manually setting the property of **InfoCoil** to 0. You can assign the value of this property either when creating the lap coil or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

An example of a whole lap winding with an orange slot core is shown in the following figure:

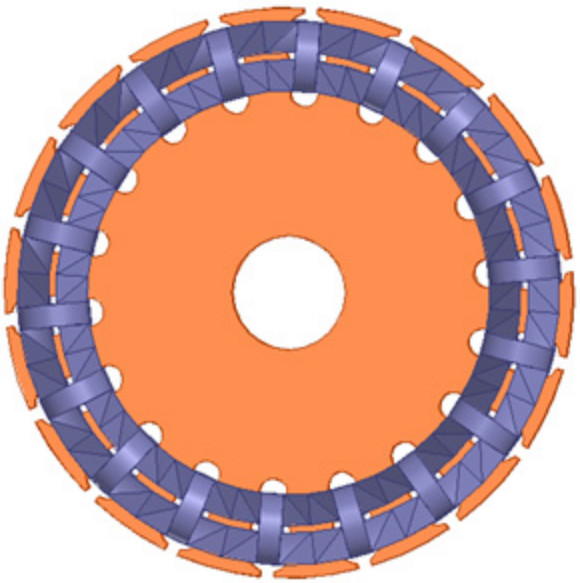


Figure 9-63 Whole Lap Winding

Creating a Single Lap Coil and its Terminals

1. Create a single lap coil by manually setting the property of **InfoCoil** to 1. Figure 9-57 ,Figure 9-58 , and Figure 9-59 show the single lap coil from different angles.
2. For the terminals of the lap coil, set the property of **InfoCoil** to 2 or 3 (terminal1 or terminal2). Figure 9-64 shows terminal1 in the coordinate system.

You can assign the value of this property either when creating the lap coil or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

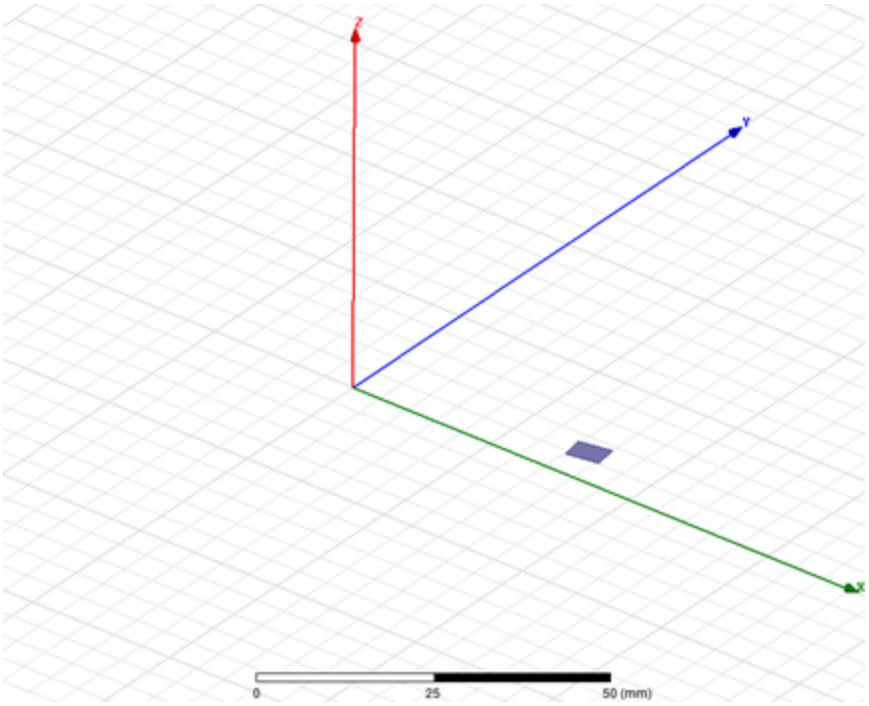


Figure 9-64 Terminal 1 of LapCoil

Creating a 3D Open Insulation

A 3D open insulation is created by manually setting the property of **InfoCoil** to 4. In the history tree, the object changes from a solid to a sheet.

You can assign the value of this property either when creating the lap coil or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

A 3D open insulation is shown in Figure 9-65 . Note that it is a sheet instead of a solid: it is empty if you observe it carefully.

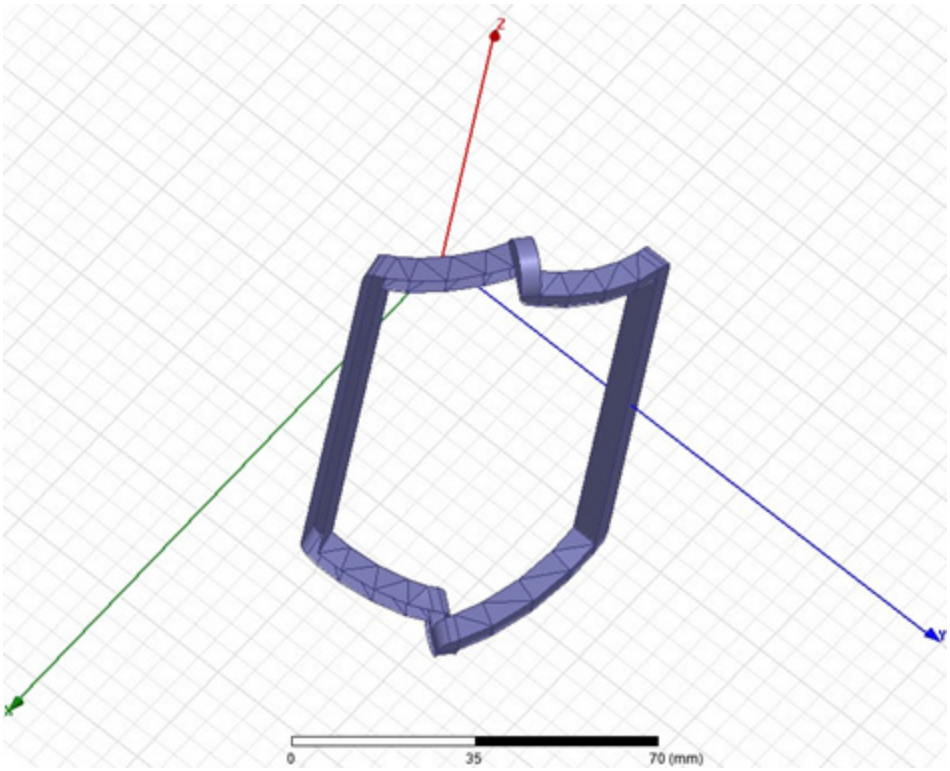
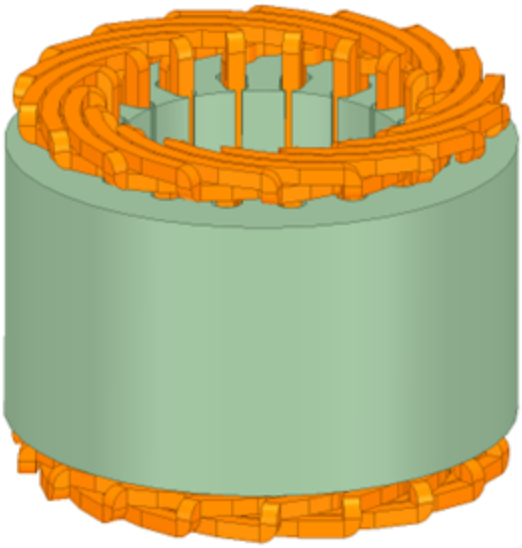


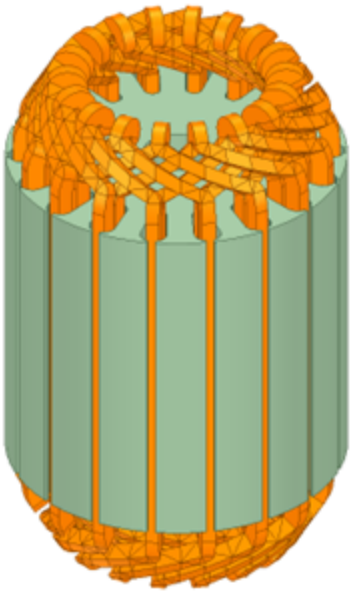
Figure 9-65 3D open insulation of LapCoil

Creating a Bended Coil

The following figures show an outer core lap winding with a 90° bended end coil (**BendAngle** = 90°), and an inner-core lap winding with a 30° bended end coil (**BendAngle** = 30°).



Outer-core lap winding with 90° bended end coil.

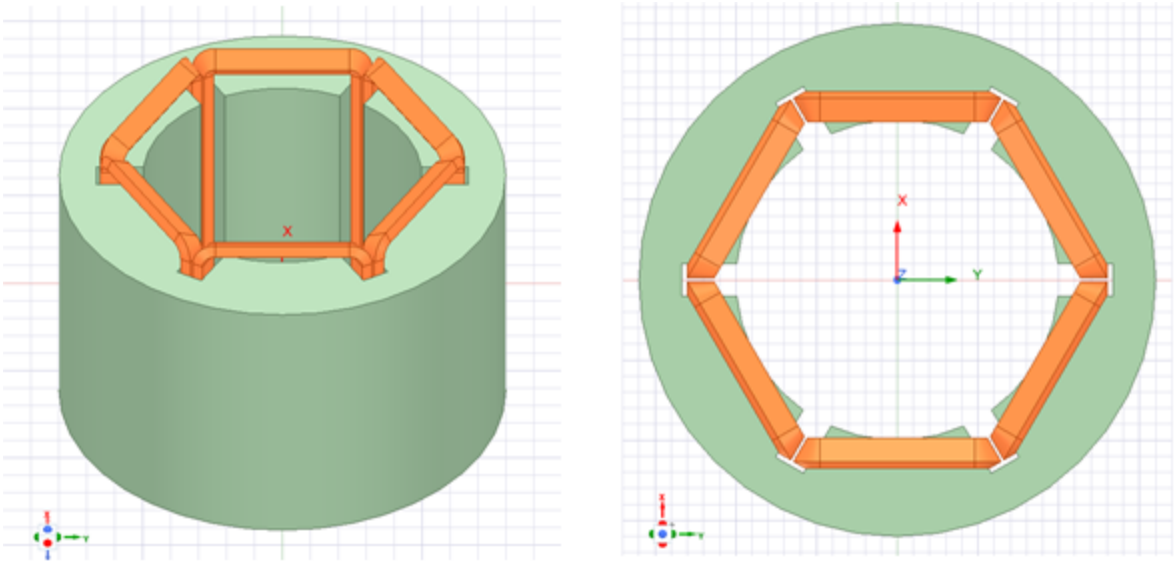


Inner-core lap winding with 30° bended end coil.

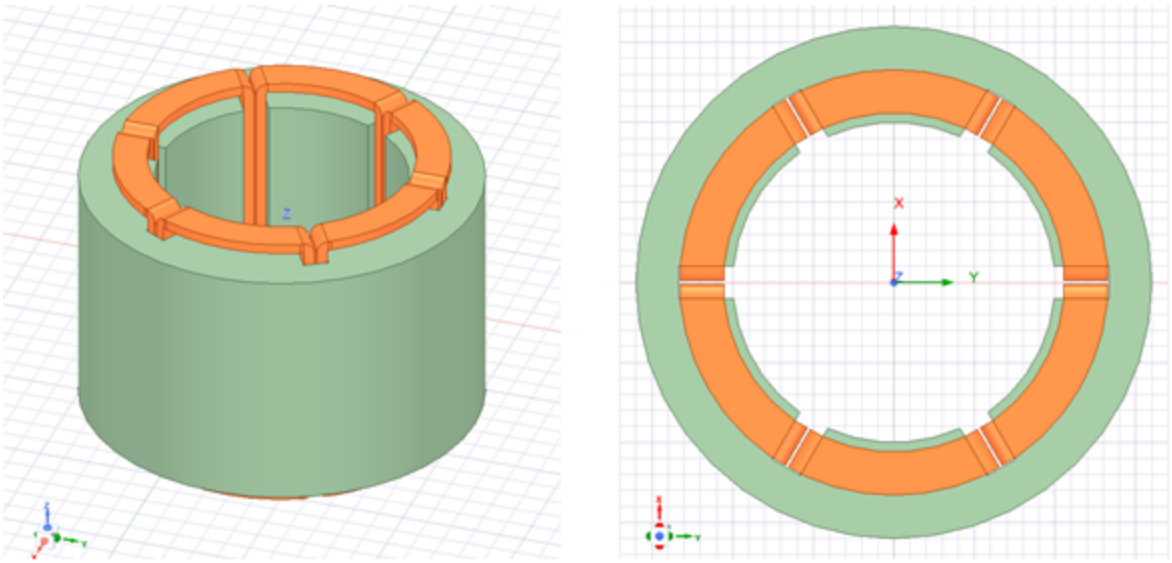
Creating a Tooth Coil

A Tooth Coil – a coil directly wound on a tooth – is a special lap coil with abs(**CoilPitch**) = 1. Examples of a tooth-coil winding with sloped straight end coil (**CoilPitch** = 1), and with flat arced end coil (**CoilPitch** = -1), are shown below. For an outer core with few slots, the straight end coil

might cover the inner bore when CoilPitch = 1. In such a case, the conductor height inside slots should be reduced to ensure that the end coil will not overlap with the inner core which might be longer than the outer core. However, if you create a tooth coil with a flat arced end coil, you do not need to reduce the conductor height inside slots.



Tooth-coil winding with sloped straight end coil (**CoilPitch** = 1).



Tooth-coil winding with flat arced end coil (**CoilPitch** = -1).

ConCoil UDP

The ConCoil UDP is used to create a coil for concentric distributed AC windings or concentric DC field windings in a slotted core created by the SlotCore, VentSlotCore, RacetrackSlotCore, or NonSalientPoleCore UDP. This UDP can also create a coil cross-section used as a terminal for current assignment.

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores.
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores.
Length	Core length.
Skew	Skew angle in core length range.
Slots	Number of slots.
SlotType	Slot Type: 1 to 7.
Hs0	Slot opening height.
Hs1	Slot wedge height.
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.
FilletType	0: a quarter circle; 1: a tangent connection.
LayerLoc	0: whole; 1: middle; 2: top; 3: bottom.
CoilPitch	Coil pitch measured in slots.
EndExt	One-side end extended length.
LayerExt	Span layer extension in the axial direction.
LayerDiff	Span layer difference in the radial direction.
AltEnd	0: same end layers; 1: alternate end layers.
LenRegion	Region length.
InfoCoil	0: coil; 1 terminal1; 2: terminal2; 100: region.

The following four 2D figures show the parameter **LayerLoc** with different values. Slot cores are used as references to help better understand concoils.

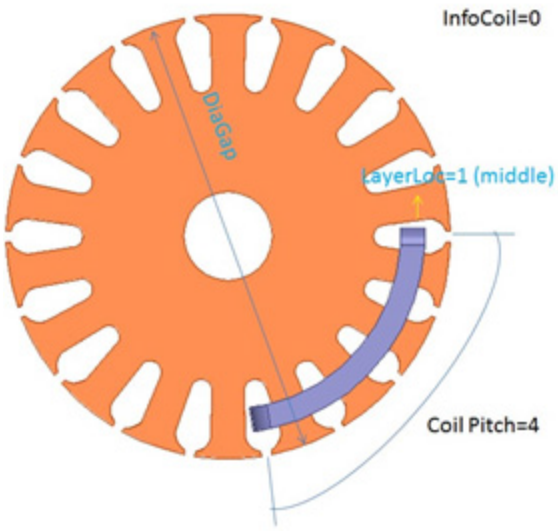


Figure 9-66 LayerLoc set to 1

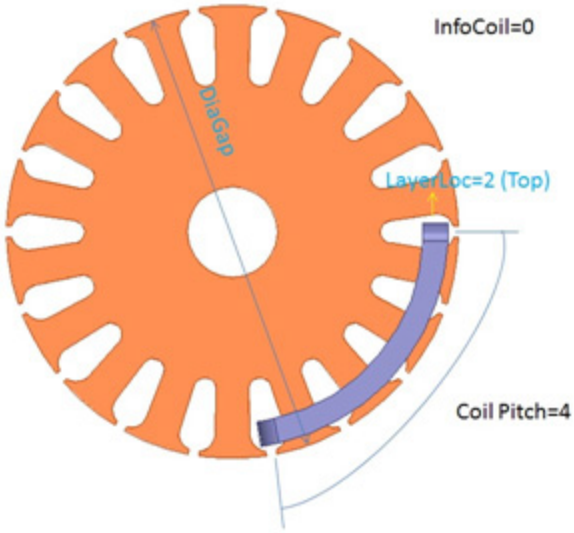


Figure 9-67 LayerLoc set to 2

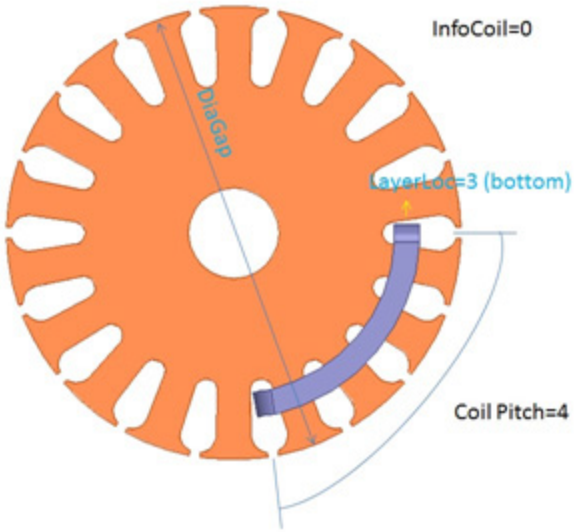


Figure 9-68 LayerLoc set to 3

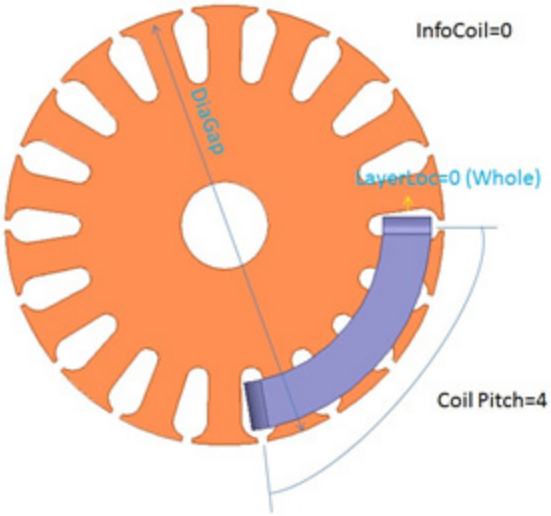


Figure 9-69 LayerLoc set to 0

The following four 3D figures illustrate parameters of ConCoil such as **LayerExt**, **LayerDiff** and **AltEnd**.

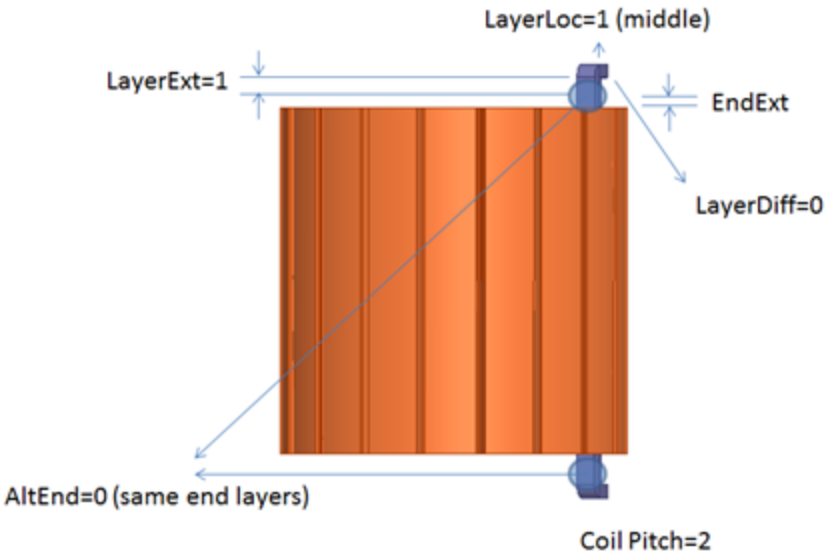


Figure 9-70 Reference 3D ConCoil

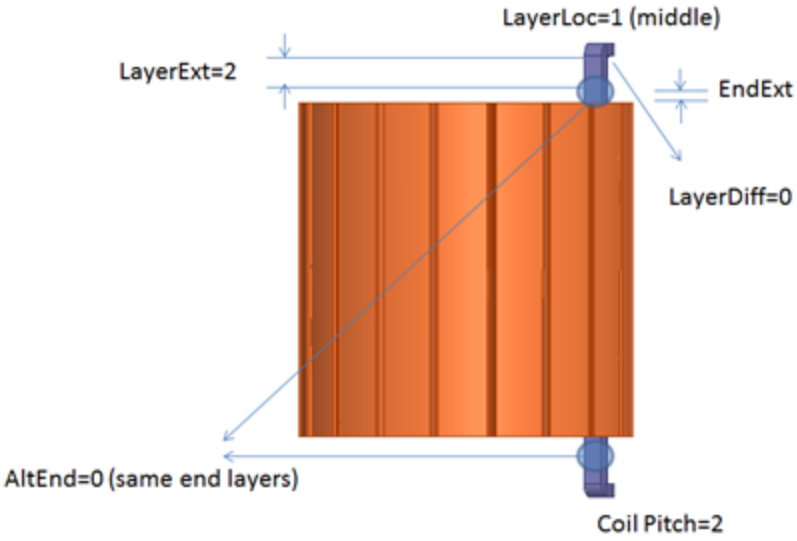


Figure 9-71 3D ConCoil with a different LayerExt than shown in Figure 9-70 .

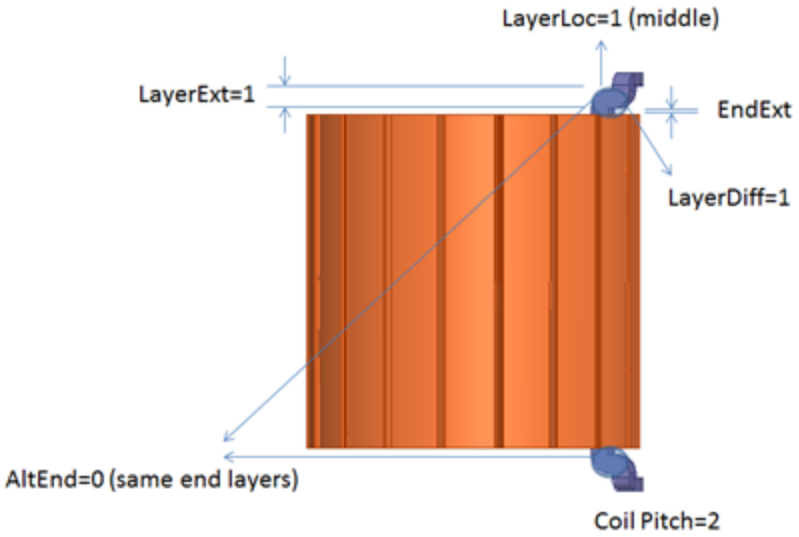


Figure 9-72 3D ConCoil with a different LayerDiff than shown in Figure 9-70 .

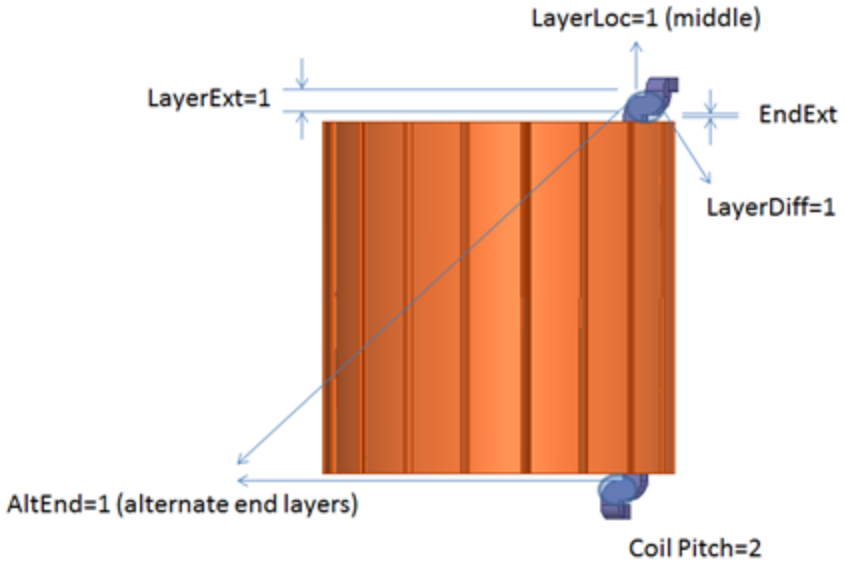


Figure 9-73 3D ConCoil with a different AltEnd than shown in Figure 9-72 .

Note	For parameters such as **Skew**, **FilletType** and parameters related to slots, refer to SlotCore UDP parameters.

Creating a Single Concentric Coil and its Terminals

You create a single concentric coil by manually setting the property of **InfoCoil** to 0.Figure 9-66 through Figure 9-70 show the single concentric coil.

For the terminals of a concentric coil, set the property of **InfoCoil** to 1 or 2 (terminal1 or terminal2). Figure 9-72 shows terminal 1 of a concentric coil.

You can assign the value of this property either when creating a concentric coil or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

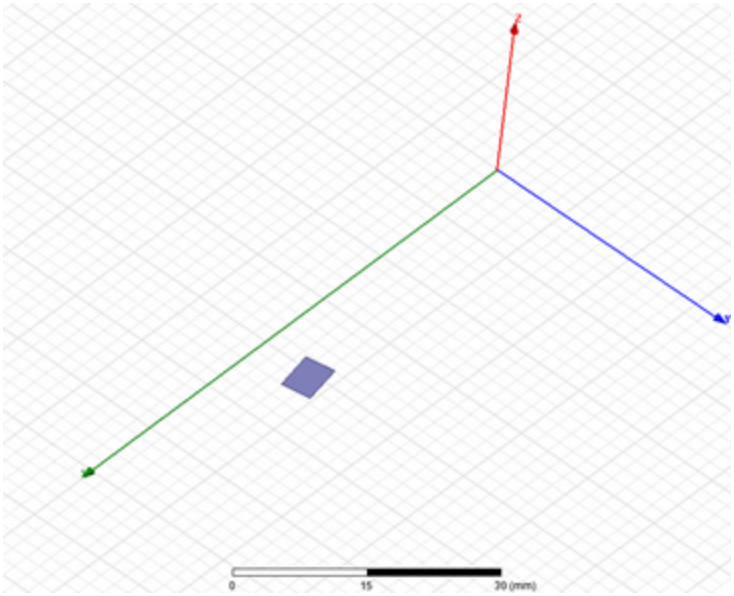


Figure 9-74 Terminal 1 of ConCoil

WaveCoil UDP

The WaveCoil UDP is used to create a wave-type coil in a slotted core created by the SlotCore or VentSlotCore UDP.

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores.
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores.
Length	Core length.
Skew	Skew angle in core length range.
Slots	Number of slots.

Property	Description
SlotType	Slot Type: 1 to 7.
Hs0	Slot opening height.
Hs1	Slot wedge height.
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.
FilletType	0: a quarter circle; 1: a tangent connection.
Layers	Number of winding layers.
CoilPitch	Coil pitch measured in slots.
EndExt	One-side end extended length.
SpanExt	Axial length of end span; 0 for no span.
SegAngle	Deviation angle for slot arches (5~15, <5 for true surface).
PolePitch	Pole pair pitch measured in slots.
LenRegion	Region length.
InfoCoil	0: winding; 1: one coil; 100: region.

The following three figures illustrate these parameters of wave coil.

Note	• A slot core (orange) is used as reference to better present the parameters and position of the wave coil. • Most parameters in a wave coil are similar or exactly the same as those in a lap coil except **PolePitch**. Figure 9-75 shows the definition of the parameter **PolePitch**. Figure 9-76 presents a winding wave coil that can be compared with a single wave coil in Figure 9-75 . Further, Figure 9-77 gives an overview of parameters in axial direction. • For parameters such as **Skew**, **FilletType**, **SegAngle** and parameters related to slots, refer to SlotCore UDP parameters.

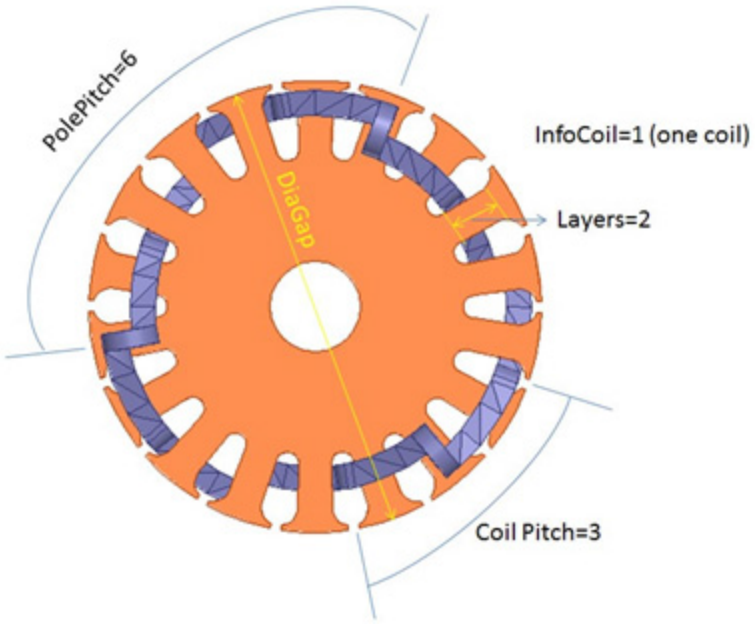


Figure 9-75 One Wave Coil

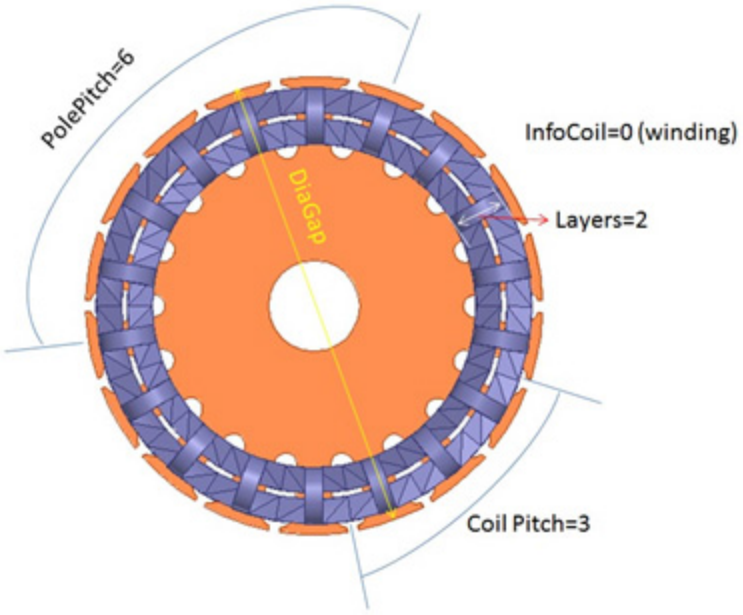


Figure 9-76 A Whole Winding Wave Coil

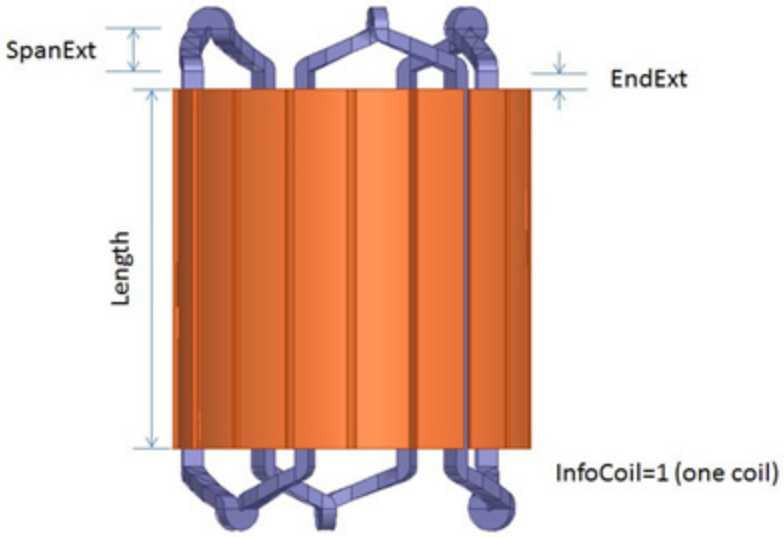


Figure 9-77 Parameters of Wave Coil in axial direction

Creating a Whole Wave Winding

You can create a whole wave winding by manually setting the property of **InfoCoil** to 0.

You can assign the value of this property either when creating a wave coil or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

An example is shown in Figure 9-76 .

Creating a Single Wave Coil

In Maxwell, wave coil is set to a single wave coil by default when being created (**InfoCoil** is equal to 1).

You can assign the value of this property either when creating a wave coil or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Examples are shown in Figure 9-75 and Figure 9-77 .

SquirrelCage UDP

The SquirrelCage UDP is used to create a squirrel-cage winding in a core created by the SlotCore or VentSlotCore UDP for single-phase or multi-phase induction machines..

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores.

Property	Description
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores.
Length	Core length.
Skew	Skew angle in core length range.
Slots	Number of slots.
SlotType	Slot Type: 1 to 7.
Hs0	Slot opening height.
Hs01	Slot closed bridge height.
Hs1	Slot wedge height.
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.
FilletType	0: a quarter circle; 1: a tangent connection.
HalfSlot	0: symmetric slots; 1: half slots.
BarEndExt	One-side bar end extended Length.
RingLength	One-side axial ring length.
RingHeight	Radial ring height.
RingDiaGap	Ring diameter on gap side.
CastRotor	0: insert-bar; 1: cast-rotor.
SegAngle	Deviation angle for slot arches (10~30, <10 for true surface).
LenRegion	Region length.
InfoCoil	0: bars & rings; 1: bars; 2: rings; 100: region.

These parameters are used in the following figures:

Note	
	1. **RingDiaGap** is set to 95 mm whereas **DiaGap** is 100 mm to distinguish these two parameters in the following figures. 2. In the following 3D figure, a slot core is drawn to show the **BarEndExt** and **Length** of Squirrel Cage. 3. For parameters such as **Skew**, **FilletType**, **SegAngle** and parameters related to slots, refer to SlotCore UDP parameters.

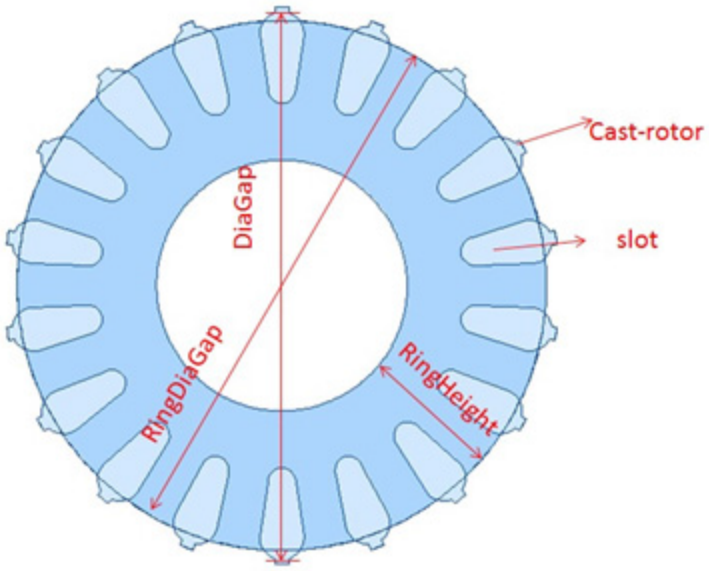


Figure 9-78 Parameters of squirrel cage on cross-section

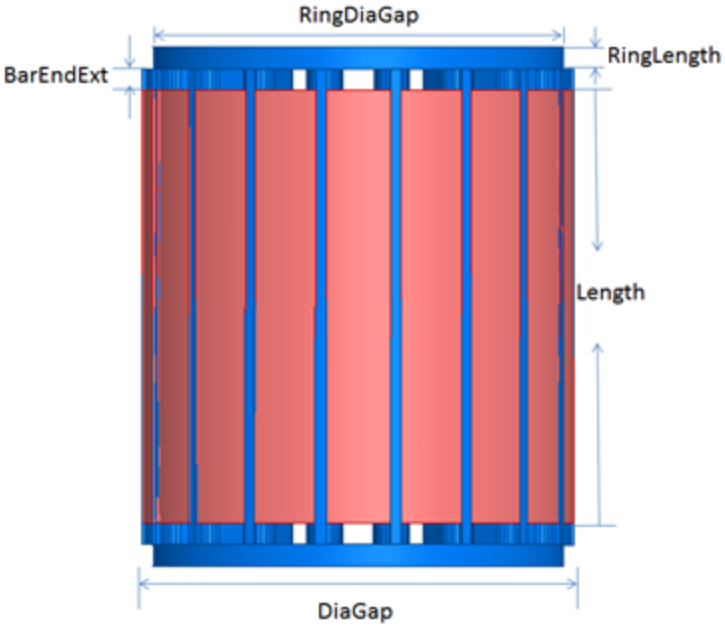


Figure 9-79 Parameters of squirrel cage on axial direction

Creating a Whole Squirrel-Cage Winding

A whole squirrel cage winding is set by default when being created (**InfoCoil** is equal to 0). An example is shown in Figure 9-79 as a blue object.

You can assign the value of this property either when creating a squirrel cage or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Creating Bars Only (Squirrel Cage)

You can create bars of a squirrel cage by manually setting the property of **InfoCoil** to 1. Figure 9-80 shows bars of a squirrel cage.

You can assign the value of this property either when creating a squirrel cage or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

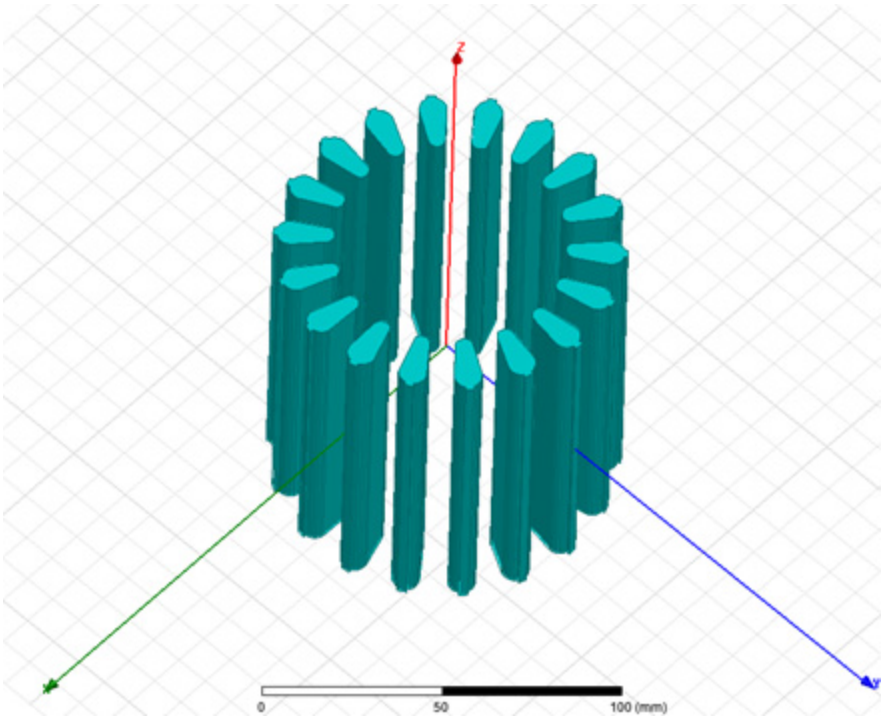


Figure 9-80 Bars of squirrel cage

Creating Two Rings Only

You can create two rings of a squirrel cage by manually setting the property of **InfoCoil** to 2. Figure 9-81 shows rings of a squirrel cage.

You can assign the value of this property either when creating a squirrel cage or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

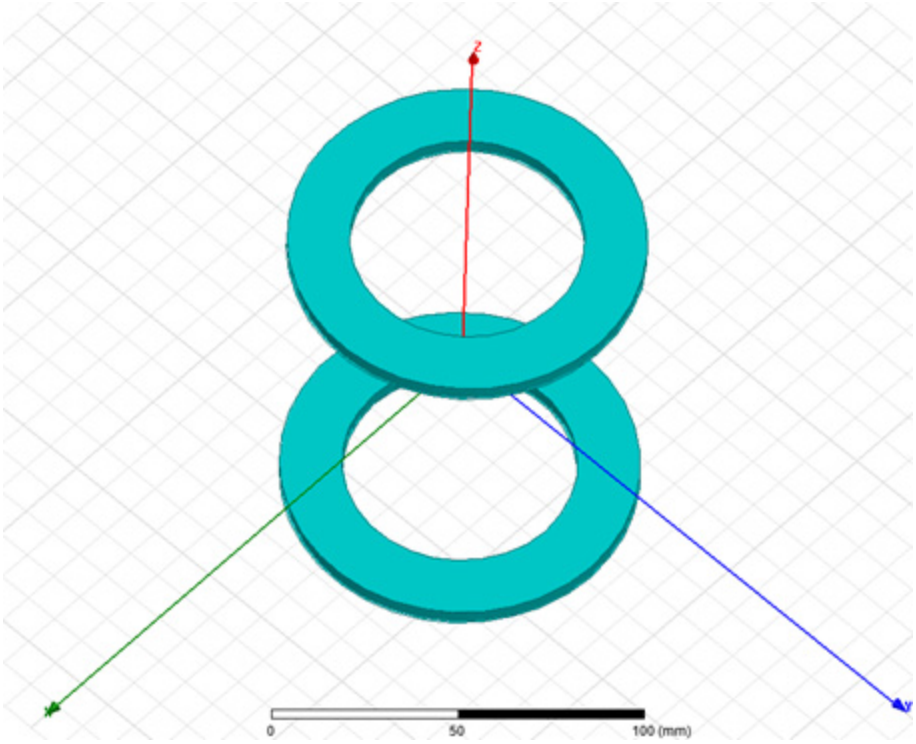


Figure 9-81 Rings of squirrel cage

DoubleCage UDP

The DoubleCage UDP is used to create a double squirrel-cage winding in a slotted core created by the SlotCore UDP or VentSlotCore UDP for induction machines with a double squirrel-cage rotor. The created double squirrel-cage winding can also be used as a subtract tool to derive a core with top and bottom slots.

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores.
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores.
Length	Core length.
Skew	Skew angle in core length range.
Slots	Number of slots.
SlotType	Slot Type: 1 to 7.
Hs0	Slot opening height.

Property	Description
Hs01	Slot closed bridge height.
Hs1	Slot wedge height.
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.
FilletType	0: a quarter circle; 1: a tangent connection.
HalfSlot	0: symmetric slots; 1: half slots.
BarEndExt	One-side bar end extended Length.
RingLength	One-side axial ring length.
RingHeight	Radial ring height.
RingDiaGap	Ring diameter on gap side.
DoubleCage	0: normal squirrel cage; 1: double squirrel cage.
BSlotType	Bottom slot type: 1 to 4.
BHs0	Slot opening height.
BHs1	Slot wedge height.
BHs2	Slot body height.
BBs0	Slot opening width.
BBs1	Slot wedge maximum width.
BBs2	Slot body bottom width, 0 for parallel teeth.
BRs	Slot body bottom fillet.
CastRotor	0: insert-bar; 1: cast-rotor.
SegAngle	Deviation angle for slot arches (10~30, <10 for true surface).
LenRegion	Region length.
InfoCoil	0: bars & rings; 1: bars; 2: rings; 100: region.

The main difference between double cage and squirrel cage is that a double cage has double slots instead of single slots. The following 2D figure shows the bottom slot, which is connected to the normal slot. The parameters for a bottom slot are similar to those of a normal slot. Other parameters are exactly the same as a squirrel cage.

Note	For parameters such as **Skew**, **FilletType**, **SegAngle** and parameters related to slots, refer to SlotCore UDP parameters.

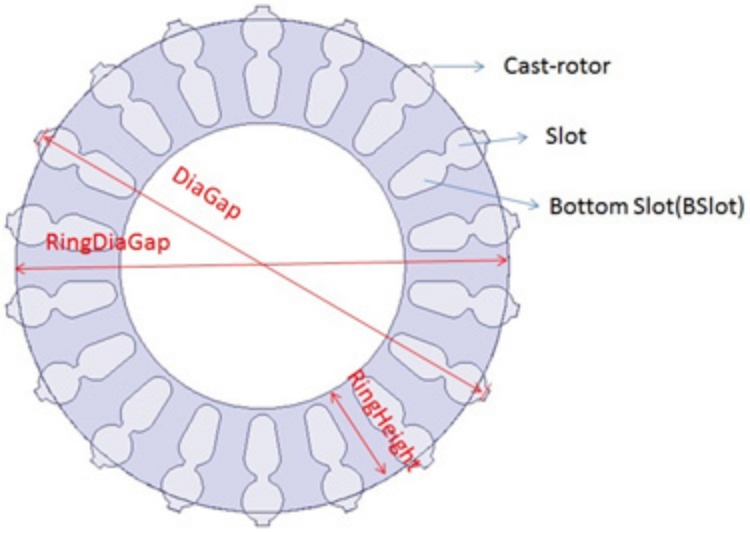


Figure 9-82 Parameters of double cage on cross-section

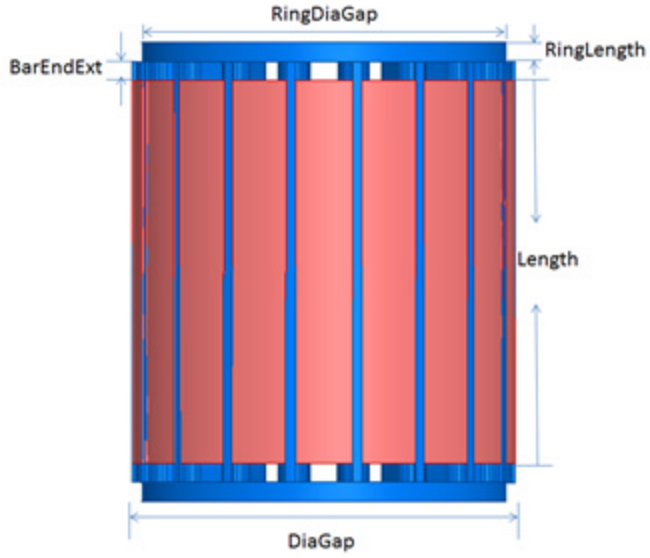


Figure 9-83 Parameters of double cage on axial direction

Creating a Whole Double-Cage Winding

A whole double-cage winding is set by default when being created (**InfoCoil** is equal to 0). An example is shown in Figure 9-83 as a blue object.

You can assign the value of this property either when creating a squirrel cage or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Creating Bars Only (Double Cage)

You can create bars of a double cage by manually setting the property of **InfoCoil** to 1. Figure 9-84 shows bars of a double cage.

You can assign the value of this property either when creating a double cage or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

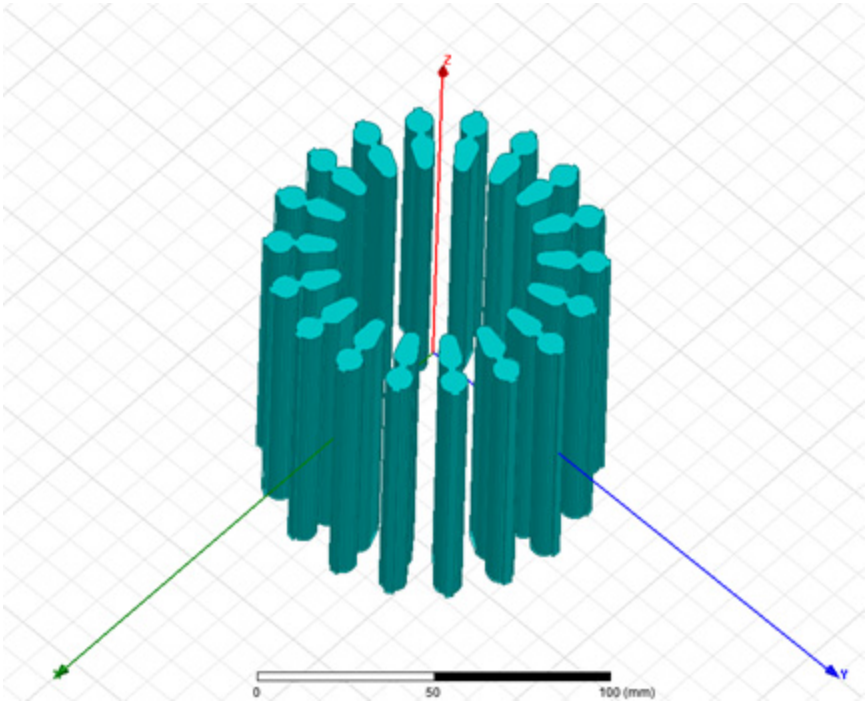


Figure 9-84 Bars of double cage

Creating Two Rings Only (Double Cage)

You can create two rings of a double cage manually by setting the property of **InfoCoil** to 2. The following figure shows rings of a double cage.

You can assign the value of this property either when creating a double cage or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

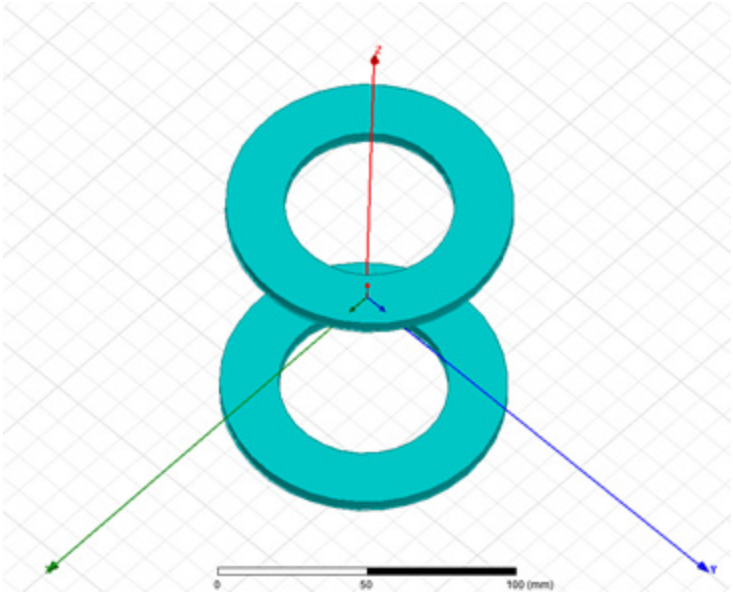
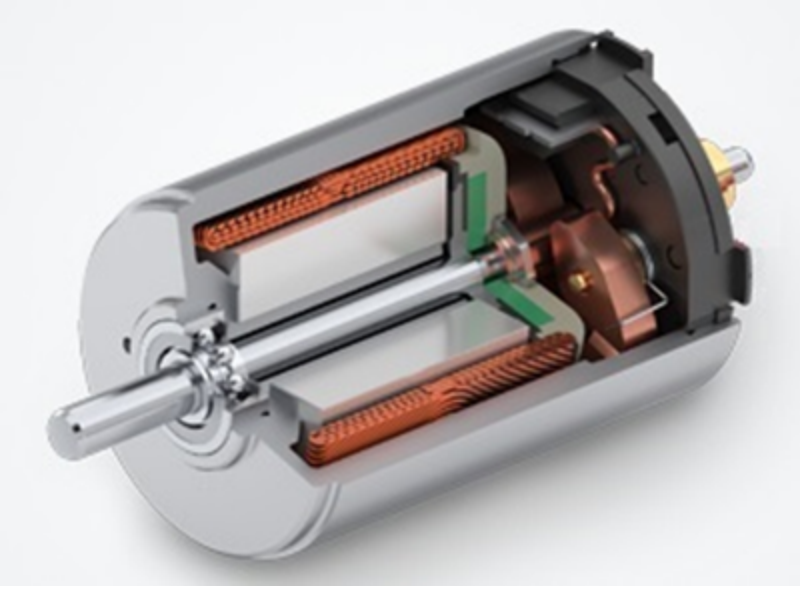


Figure 9-85 Rings of double cage

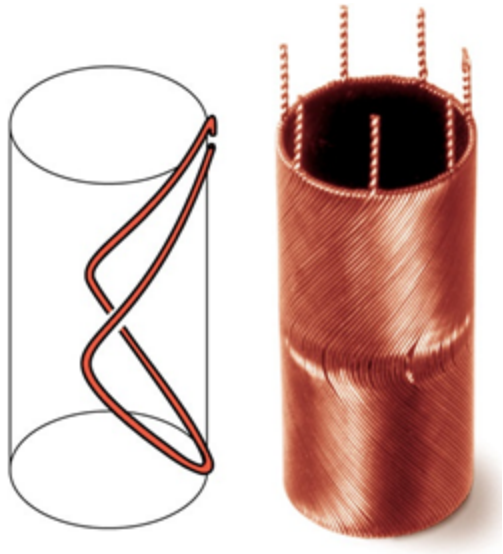
CupCoil UDP

The CupCoil UDP is used to create a coil for distributed slotless AC windings or coreless DC commutating windings in the air-gap between outer and inner cores. This UDP can also create a coil cross-section, which can be used as a terminal for current assignment.

Property	Description
DiaOuter	Coil outer diameter
DiaInner	Coil inner diameter
Length	Length of coil straight part
Coils	Number of total coils in a circle
CoilType	1: lap; 2: wave
CoilPitch	Coil pitch in degrees
LenTotal	Tip-to-tip length of a coil, 0 for 2D
LenRegion	Region length
InfoCoil	0: coils; 1: coil; 2 terminal1; 3: terminal2; 100: region

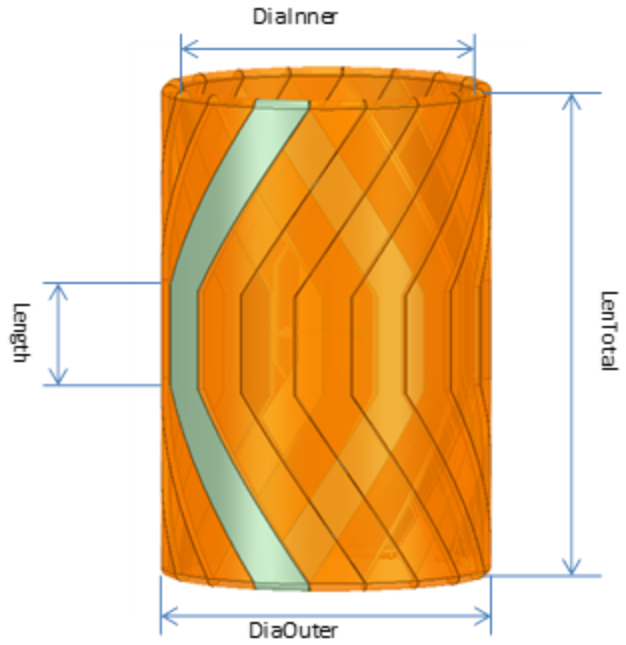


Coreless DC motor



Coreless DC Winding

The following figure shows some of the parameters described in the table above. To avoid too small a mesh size, wire turns are not created one-by-one as solid conductors. Instead, all adjacent wires carrying the same current are grouped as a strand coil with one or more turns.

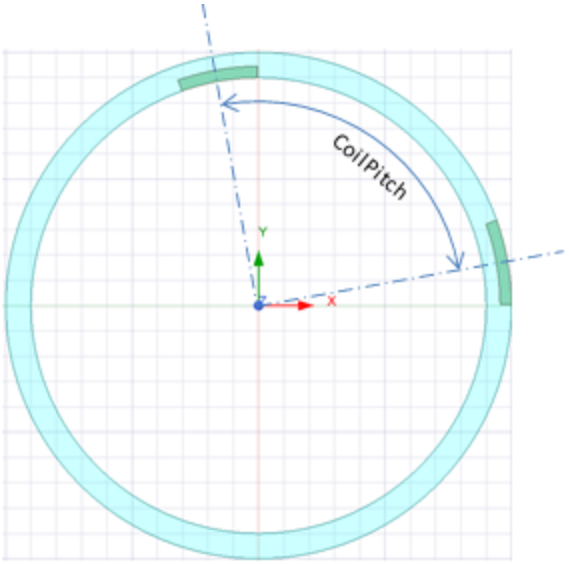


A coreless lap-type DC winding created by CupCoil UDP

Creating a Coil and its Terminals (CupCoil)

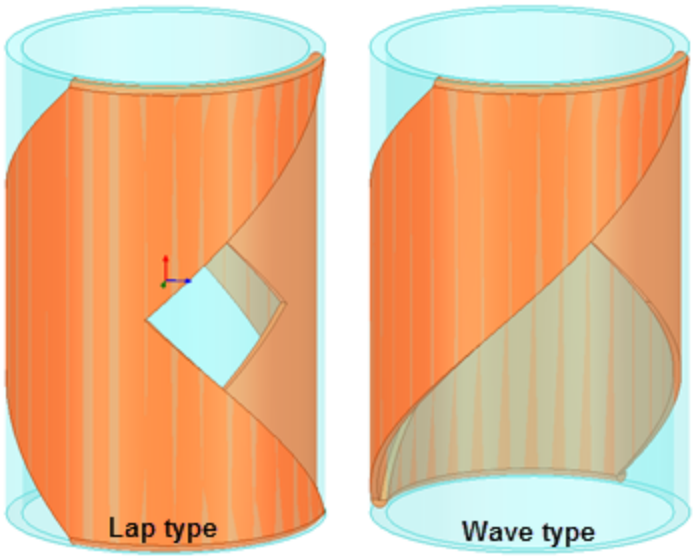
When **InfoCoil** = 0, the created object includes all coils which are united together. To separate them you can use **Modeler>Boolean>Separate Bodies** menu command. When **InfoCoil** = 1, the created object includes only one coil, as shown by the coil in green color in the figure. You can use menu command **Edit>Duplicate>Around Axis** to duplicate it.

When **InfoCoil** = 2, you can create a terminal for coil assignment. You can create a return terminal for a coil in 2D by setting **InfoCoil** = 3. The following figure shows both “go” and “return” terminals, and shows the definition of the **CoilPitch** parameter (90 degrees in this example).



“go” and “return” terminals for a CupCoil with a given CoilPitch

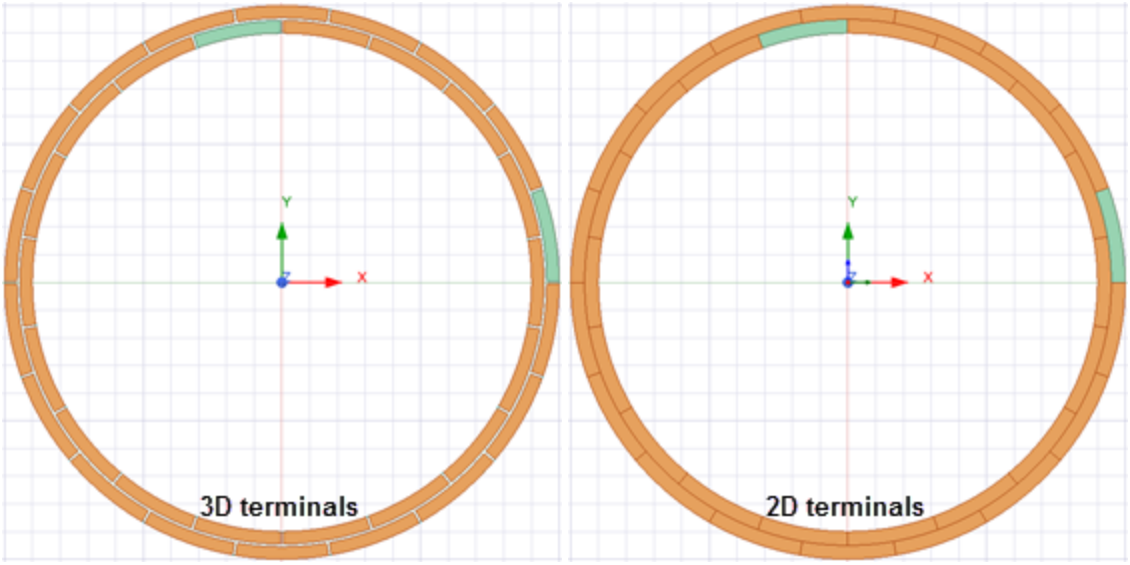
When **CoilType** = 1 , the created coil is a lap-type. When **CoilType** = 2, the created coil is a wave-type. For a wave-type coil, the bottom span will be skewed in the same direction as the top span, and therefore, the bottom tip will shift from the top tip by one pole pitch. The pole pitch will be 180°, 90°, 60°, …, for 2, 4, 6, …, poles, respectively. The input value of **CoilPitch** will be adjusted to the closest value of possible pole pitches. The following figure compares a wave-type coil with a lap-type coil.



Wave-type coil compared with a lap-type coil

You can set **LenTotal** = 0 to create coils or terminals in 2D, or directly use **InfoCoil** = 2 and **InfoCoil** = 3 with **LenTotal** > 0 to draw 3D terminals for 2D coils. The difference between 2D

terminals and 3D terminals is that: there will be a gap between any two adjacent 3D coil terminals, but there will be no gap between 2D terminals to improve mesh quality, as compared in the figure below.



2D terminals compared with 3D terminals

A user needs to input an average CoilPitch when we try to create coil terminals for 2D FEA model. For a wave-type coil, since the bottom span is continuously skewed without changing the skew direction from the top span, the average coil pitch is the same as the real CoilPitch. For a lap-type coil, the average coil pitch can be derived from:

PitchAve = (Length x CoilPitch + (LenCore – Length) x (CoilPitch + PitchMin) / 2) / LenCore

where **LenCore** is the length of the inner and outer cores which carry magnetic field, and

PitchMin = (LenTotal – 2 x RadTip – LenCore) / (LenTotal – 2 x RadTip – Length) x CoilPitch

which must be limited within (0, CoilPitch). In the equation above, **RadTip** is the radius of top and bottom coil tips, which is obtained from:

RadTip = (DiaOuter – DiaInner) / 4

The skew angle for multi-slice 2D FEA model can be obtained from:

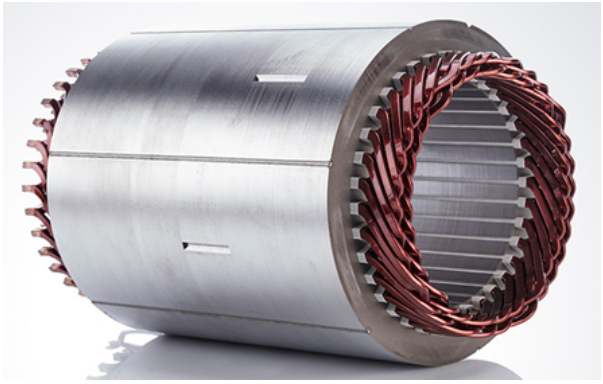
AngSkew = (PitchAve – PitchMin) x 2

which is applicable for both lap-type and wave-type coils.

HairpinCoil UDP

The HairpinCoil UDP is used to create a hairpin-type coil. Hairpin coils are used to make a hairpin winding in a slotted core. All hairpin coils belonging to the same phase are welded in series to perform a hairpin winding.

Hairpin Coil



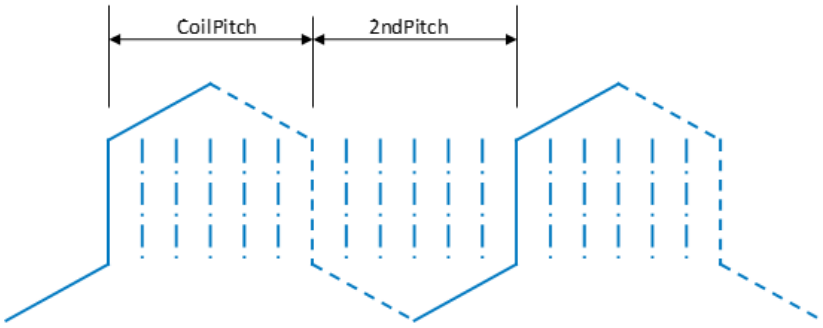
Stator with Hairpin Winding

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores.
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores.
Length	Core length, 0 for 2D.
Skew	Skew angle in core length range.
Slots	Number of slots.
SlotType	Slot Type: 1 to 7.
Hs0	Slot opening height.
Hs1	Slot wedge height.
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.
FilletType	0: a quarter circle; 1: a tangent connection; 2&3: arc bottom; 4&5: V bottom
Layers	Number of total coil layers in a slot, should be an even number.
CoilPitch	Coil pitch measured in slots.
EndExt	One-side end extended length.
SpanExt	Axial length of end span; 0 for no span.
SegAngle	Deviation angle for slot arches (5~15, <5 for true surface).
2ndPitch	The second coil pitch, the pitch between two conductor sides to be welded
2ndSpanExt	Pole pair pitch measured in slots.

Property	Description
WireWidth	Wire width in slot width direction.
WireThick	Wire thickness in slot depth direction.
WireRad	Wire fillet radius, used to calculate equivalent wire size based on wire area.
LayerSetGap	Gap between two layer-sets, one layer-set is two layers of one coil.
LayerSetIdx	Layer-set index for the location of the coil to be drawn.
LayerGap	Gap between two layers in a layer set.
WindingType	1: whole-coiled (2 coil sets per pole pair); 2: half-coiled (1 coil set per pole pair).
CoilSetIdx	0: create 1st coil set; 1: create the 2nd coil set (for whole-coiled winding only).
LenRegion	Region length, to limit lead length for open coils.
InfoCoil	0: winding; 1: one coil; 100: region.

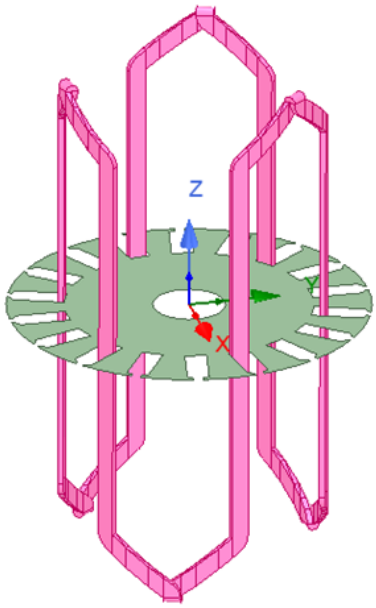
NOTE:

- Parameter **Layers** defines the total number of layers, or total number of conductors per slot. The value must be an even number. Two layers related to one coil are referred as one layer-set.
- Parameter **2ndPitch** which defines the second coil pitch (shown below) is the pitch between two coil sides to be welded.



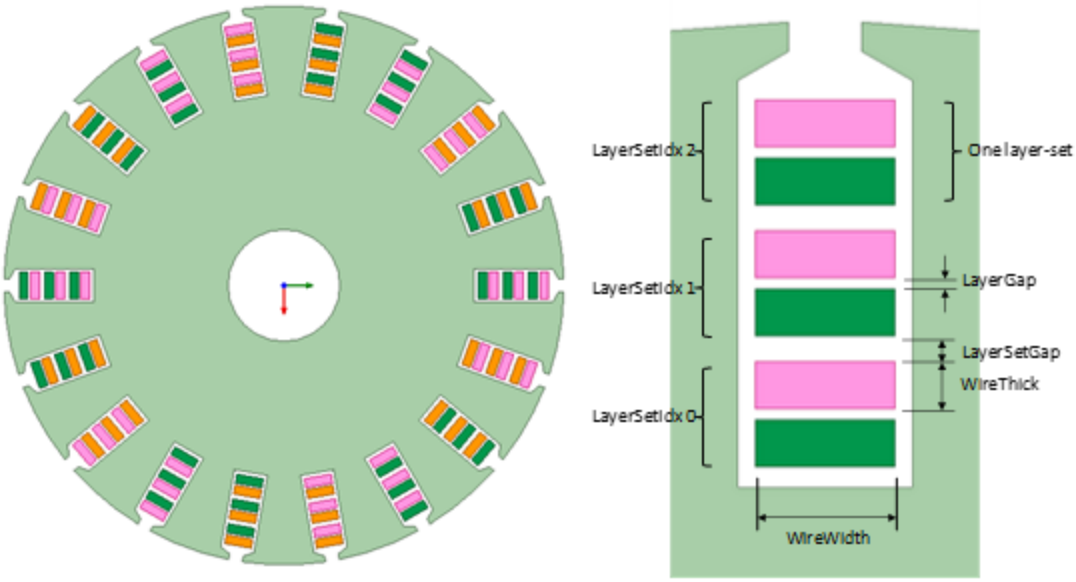
Parameters for CoilPitch and 2ndPitch

For an integer-slot winding, **Slots** value is a multiple of **2ndPitch** plus **CoilPitch** values. In such a case, the last coil in a coil chain will close the first coil, resulting in a closed coil chain, as shownbelow.



Closed hairpin-coil chain with 2ndPitch = 3 in an 18-slot core lamination

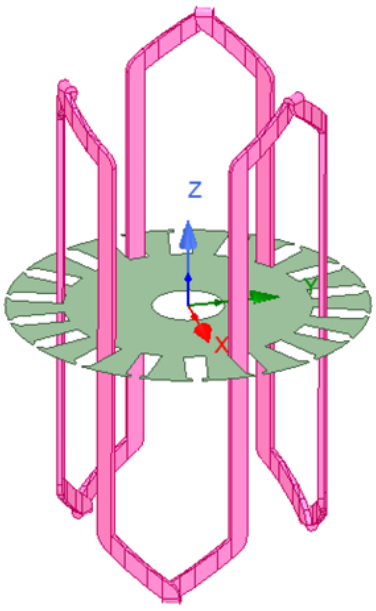
- In order to obtain good mesh quality, the UDP will not create wire with wire fillet radius, as shown above, but the effect of wire fillet radius on the wire area will be considered using a scaling factor for both **WireWidth** and **WireThick**. The scaling factor k = (WireArea / WireWidth / WidthThick) ^ 0.5, where WireArea is the wire cross-section area considering wire fillet.
- The UDP will get the maximum available rectangular space inside a slot via an optimization process for wire arrangement. The minimum gap between a wire and a rectangle side will be default as 0.3mm, considered as the thickness of a slot liner insulation. The gap between two layers in one layer-set is the same for all layer-sets. The value is automatically derived from the height of the rectangular space minus total wire thickness and total layer-set gap.
- **WindingType** and **CoilSetIdx** are used for fractional-slot windings only. You do not need to input these values for integer-slot windings. For a fractional-slot winding, the last coil in a coil chain will not close the first coil, and the coil chain will continue to include all coils belong to the same coil set. Finally, the UDP will create an opened coil set. Two lead wires will extend the opened coil set to touch one region end. For whole-coiled windings, since the two coil sets may have different coil arrangement, we need to create two coil sets separately. Parameter **CoilSetIdx** indicates which coil set is being created. The coil arrangement is based on three-phase windings.



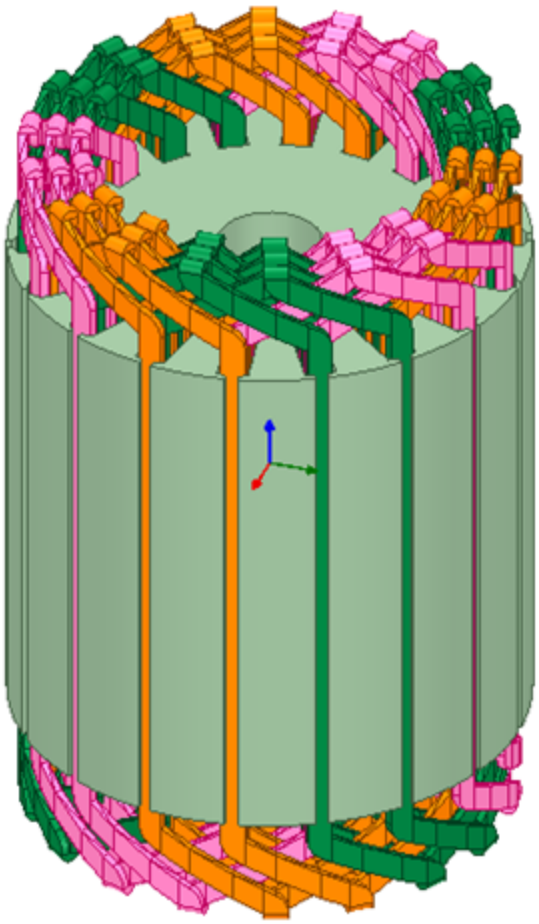
- For other parameters, please refer to *LapCoil UDP*.

Creating Closed Three-Phase Hairpin Windings (HairpinCoil)

1. In Maxwell 3D, launch HairpinCoil UDP with default setting, that is **Slots**=18, **Layers**=6, **CoilPitch**=3, **2ndPitch**=3, **LayerSetIdx**=0, and **InfoCoil**=1. Since the ratio of **Slots** to (**2ndPitch** + **CoilPitch**) is 3 (an integer number), the UDP will create a closed-coil chain, or a coil loop, located at slot bottom, as shown below (excluding the core lamination).



2. Make a copy of the coil loop, and change **LayerSetIdx** to 1, this copied coil loop is changed to locate at the slot center.
3. Make a copy of the coil loop again, and change **LayerSetIdx** to 2, this copied coil loop is changed to locate at the slot top.
4. Duplicate all these three coil loops with the following parameters in **Duplicate/Around Axis** command: **Angle** = 20deg, **Total Number** = 6, and change colors for different phases, then you will get the final three-phase windings, as shown below.



Closed three-phase hairpin windings

Creating One Phase Half-Coiled Windings (HairpinCoil)

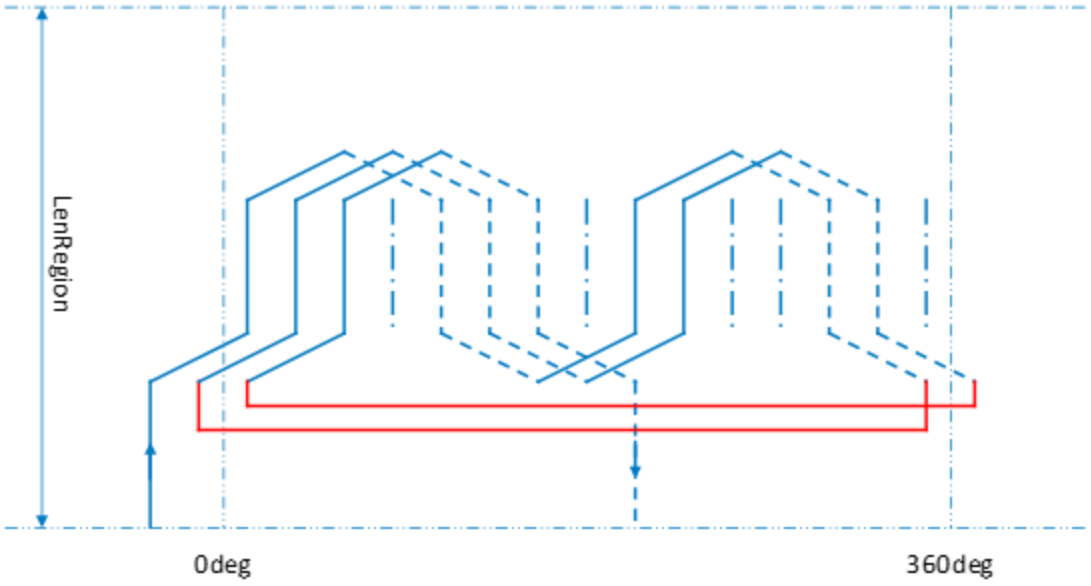
Set **Slots**=15, **Layers**=2, **CoilPitch**=4, **2ndPitch**=4, **LayerSetIdx**=0, and **InfoCoil**=1. Since the ratio of **Slots** to (**2ndPitch** + **CoilPitch**) is 15/8 (not an integer number), the UDP will create an opened coil set. In this UDP, we will support a fractional-slot winding that satisfies:

(2ndPitch + CoilPitch) x PolePairs = Slots ± 1

For this example, the number of pole pairs is 2, and the winding arrangement, for top coil side only, will be:

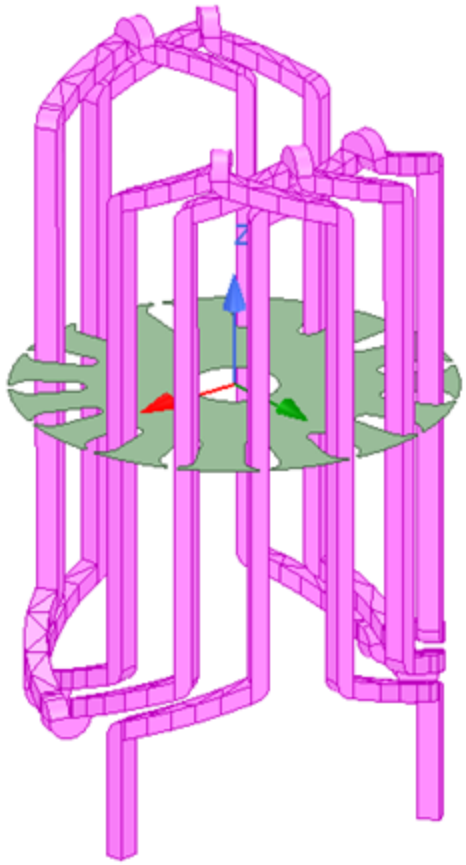
AAABBCCCAABBBCC

For this half-coiled winding, all hairpin coils labeled with “A” are included in one opened coil set, which has already contained all phase A coils, as shown below, where red lines indicate the short-circuit connection without using physical wires.



Flat expanded view of phase-A hairpin winding containing only one opened coil set

The created 3D opened winding is shown below.



3D view of phase-A hairpin winding containing only one opened coil set

Creating One Phase Whole-Coiled Windings (HairpinCoil)

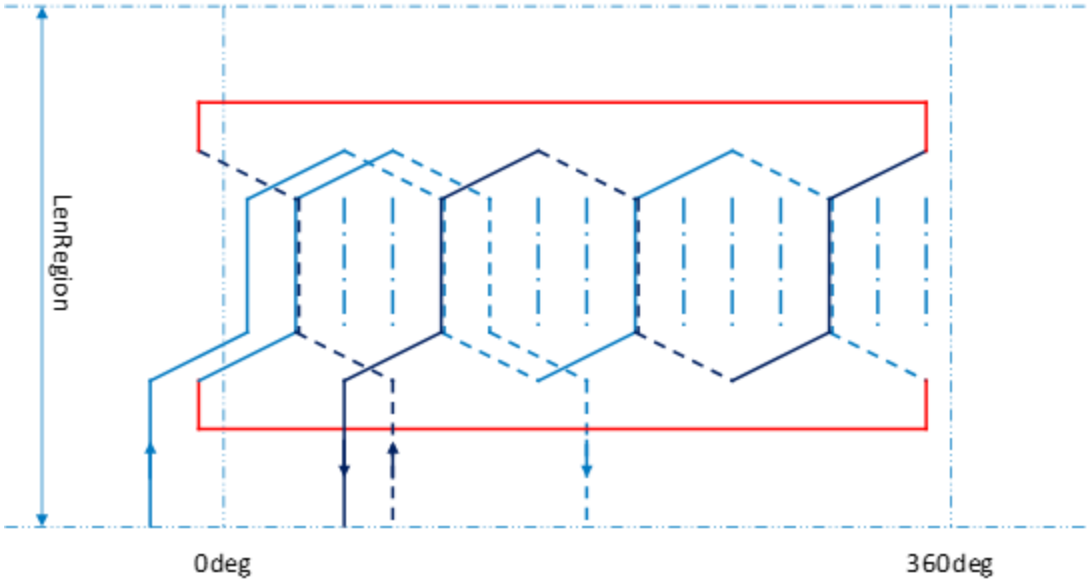
Set **Slots**=15, **Layers**=2, **CoilPitch**=4, **2ndPitch**=4, **LayerSetIdx**=0, **WindingType** = 1, **CoilSetIdx** = 0, and **InfoCoil**=1. Since the ratio of **Slots** to (**2ndPitch** + **CoilPitch**) is 15/8 (not an integer number), the UDP will create an opened coil chain. This fractional-slot winding satisfies:

(2ndPitch + CoilPitch) x PolePairs = Slots ± 1

where the number of pole pairs is 2. The winding arrangement, for top coil side only, is:

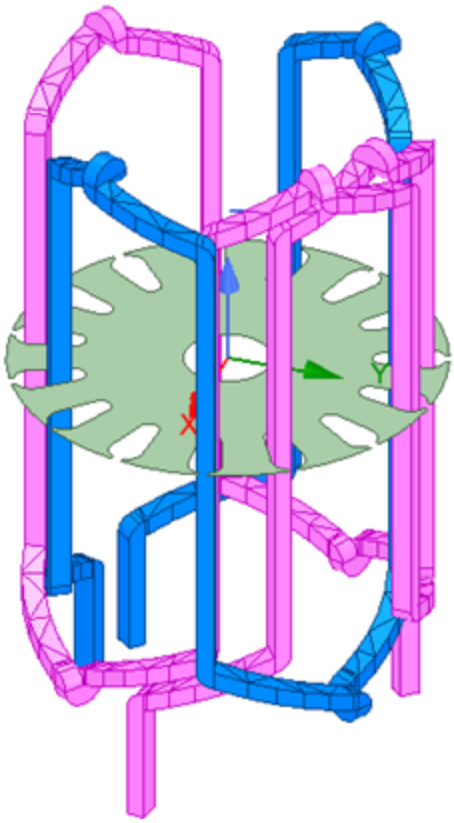
AAZBXCCYAZBBXCY

With the above parameter setting, this UDP will create an opened coil set including all coils related to all “A” coils of the above winding arrangement. Copy the coil set, and change **CoilSetIdx** to 1. The coil set is modified to include all coils related to all “X” coils of the winding arrangement. The phase-A winding is made up of these two coil sets, as shown below, where red lines indicate the short-circuit connection without using physical wires.



Flat expanded view of phase-A hairpin winding containing two opened coil sets (all coils with light blue color relate to “A” coils, and all coils with dark blue color relate to “X” coils, in the winding arrangement)

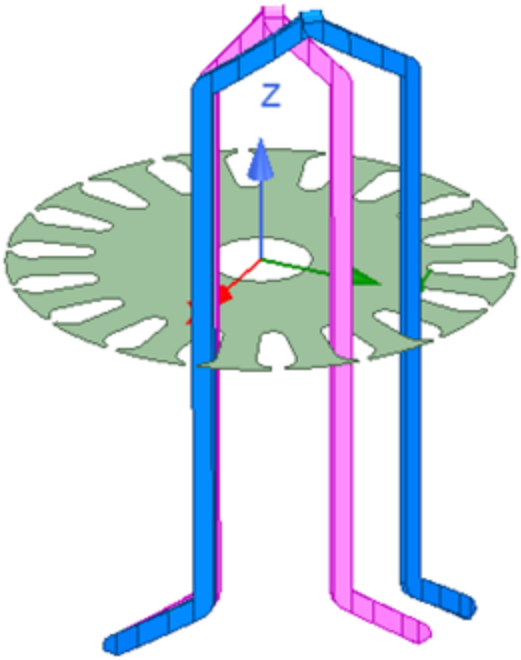
The created 3D opened winding is shown below.



3D view of phase-A hairpin winding containing two opened coil sets

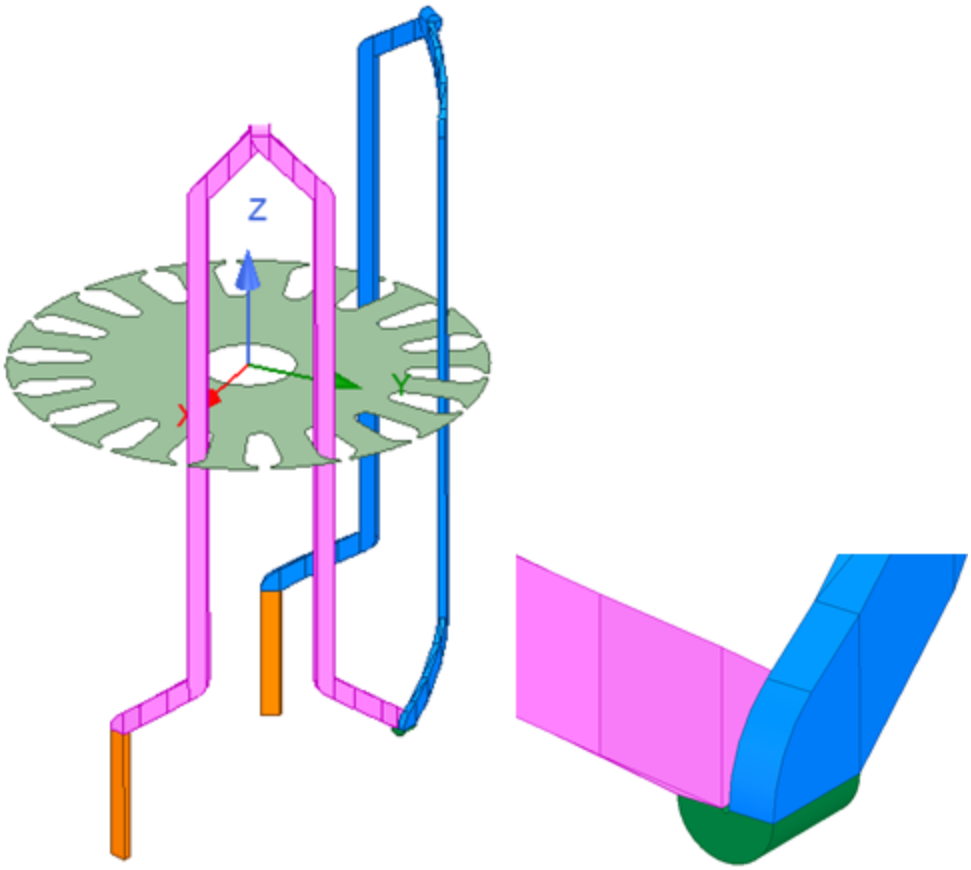
Creating Two Hairpin Coils Connected with End-Tip and Leads

1. Set **Slots**=18, **Layers**=6, **CoilPitch**=2, **2ndPitch**=3, **LayerSetIdx**=0, and **InfoCoil**=4. These parameter settings will create a single hairpin coil located at the slot bottom, as shown by the pink-color coil in the figure below.



Two hairpin coils with different coil pitch at different slot locations

2. Make a copy of the coil, and change **CoilPitch** to 3, and **LayerSetIdx** to 1. The copied coil is modified to be located at the slot center, as shown by the blue-color coil in the above figure.
3. Rotate the blue-color coil by 100deg, or 5 slot-angle, the bottom leg-span of the blue-color coil will align with the top leg-span of the pink-color coil, as shown in the figure below.



Two hairpin coils at different slot locations connected with end-tip and leads

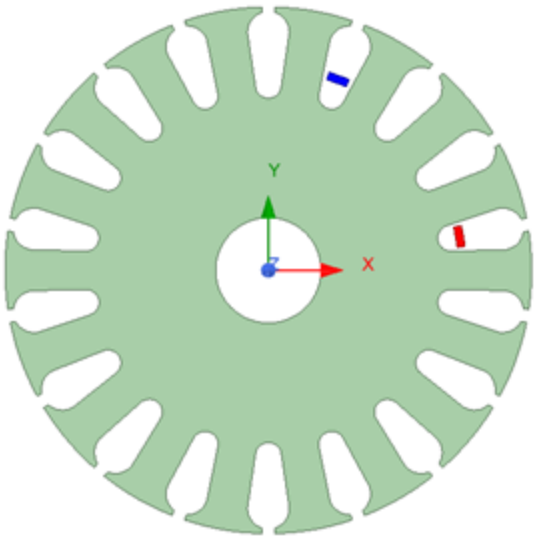
4. Make a copy of the blue-color coil, and change **LayerSetIdx** to 0.5 (for end-tip between layer-set index of 0 and 1) and **InfoCoil** to 4. The copied coil is modified to an end tip, as shown by the green-color object in the above figure.
5. Make a copy of the pink-color coil and change **InfoCoil** to 6, the copied coil is modified to a going lead; and make a copy of the blue-color coil and change **InfoCoil** to 7. The copied coil is modified to a returning lead, as shown by the golden-color objects in the figure.

The two hairpin coils, two leads, and one end-tip can thus be united to become a single connected coil set.

Note: In order for the end-tip and leads to be able to perfectly connect two hairpin coils, the parameter value of **2ndSpanExt** for all components must be the same to keep the same axial span length. If you see a message such as: “HairpinCoil - 2ndSpanExt is modified from xxx to xxx”, you need to increase the value of **2ndSpanExt** for all components to be at least the maximum modified value of all components.

Creating a Terminal for a Hairpin Coil

A terminal is a 2D cross section of a 3D coil that can be used to assign current or voltage excitations. You can create a *going* terminal by setting **InfoCoil** to 2; or you can create a *returning* terminal by setting **InfoCoil** to 3. In a connected 3D coil set, you need to create only one terminal, but you need to create both *going* and *returning* terminals for each hairpin coil for a 2D model. *Going* and *returning* terminals are shown below.



***Going* terminal (red) and *returning* terminal (blue) inside two slots**

UDPs for Pole Cores and Coils

UDPs defined for pole cores and coils include:

- SalientPoleCore UDP
- DCMCore UDP
- ShadedPoleCore UDP
- UnivMCore UDP
- SRMCore UDP
- SynRMCore UDP
- ClawPoleCore UDP
- PMCore UDP
- IPMCore UDP
- PMDamperCore UDP
- StepMCore UDP
- Square Yoke UDP
- UPMCore UDP

- VPMCore UDP
- VentCore UDP

SalientPoleCore UDP

The SalientPoleCore UDP is used to create a salient-pole core, a field winding, and a damper for salient-pole synchronous machines. It can also create a coil cross-section used as a terminal for current assignment.

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores.
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores.
Length	Core length.
Skew	Skew angle in core length range.
Slots	Number of slots.
SlotType	Slot Type: 1 to 4.
Hs0	Slot opening height.
Hs01	Slot closed bridge height.
Hs1	Slot wedge height.
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.
FilletType	0: a quarter circle; 1: a tangent connection; 2&3: arc bottom; 4&5: V bottom.
SlotPitch	Slot pitch in mechanical degrees, refer to offset point.
CenterPitch	Center slot pitch in mechanical degrees, refer to offset point.
Poles	Number of poles.
PoleType	Pole type: 1 or 2 (not valid for inner cores)
WidthShoe	Pole shoe width.
HeightShoe	Pole shoe height at pole center
FilletShoe	Pole shoe tip fillet radius.
WidthBody	Pole body width.
HeightBody	Pole body height.
AirGap2	Second air gap length.
Offset	Pole arc offset.
Off2_x	The second pole arc offset perpendicular to the pole-center line.

Property	Description
Off2_y	The second pole arc offset parallel with the pole-center line.
Bp1/Rp1	Bp1: for pole type 1; or Rp1: shoe fillet radius for pole type 2.
Bp2/Rp2	Bp2: for pole type 1; or Rp2: shoe fillet radius for pole type 2.
CoilEndExt	One-side coil end extended length.
EndRingType	0: whole press board; 1: pole press board; 2: pole ring; 3: whole ring.
BarEndExt	One-side damper bar end extended Length; for types 2 & 3 only.
RingLength	One-side axial ring length, or conductor press board thickness.
RingHeight	Radial ring height, for types 2 & 3 only.
SegAngle	Deviation angle for slot arches (10~30, <10 for true surface).
LenRegion	Region length.
InfoCore	0: core; 1: core & coils; 2: coil; 3: damper; 4: terminal1; 5: terminal2; 6:poles; 7: yoke; 100: region.

These parameters are used in the following figures. In figures with both gray and red objects, the gray objects are used as references.

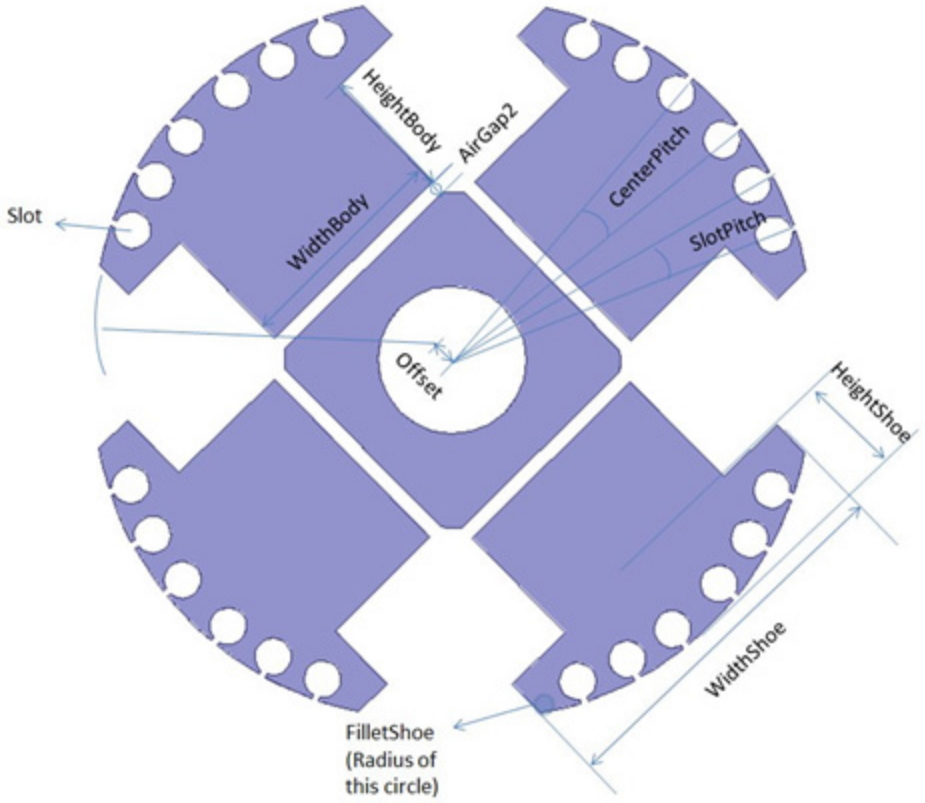


Figure 9-86 Salient Pole Core (InfoCore set to 0)

Figure 9-87 This UDP supports 2 pole types for outer cores, as shown below;

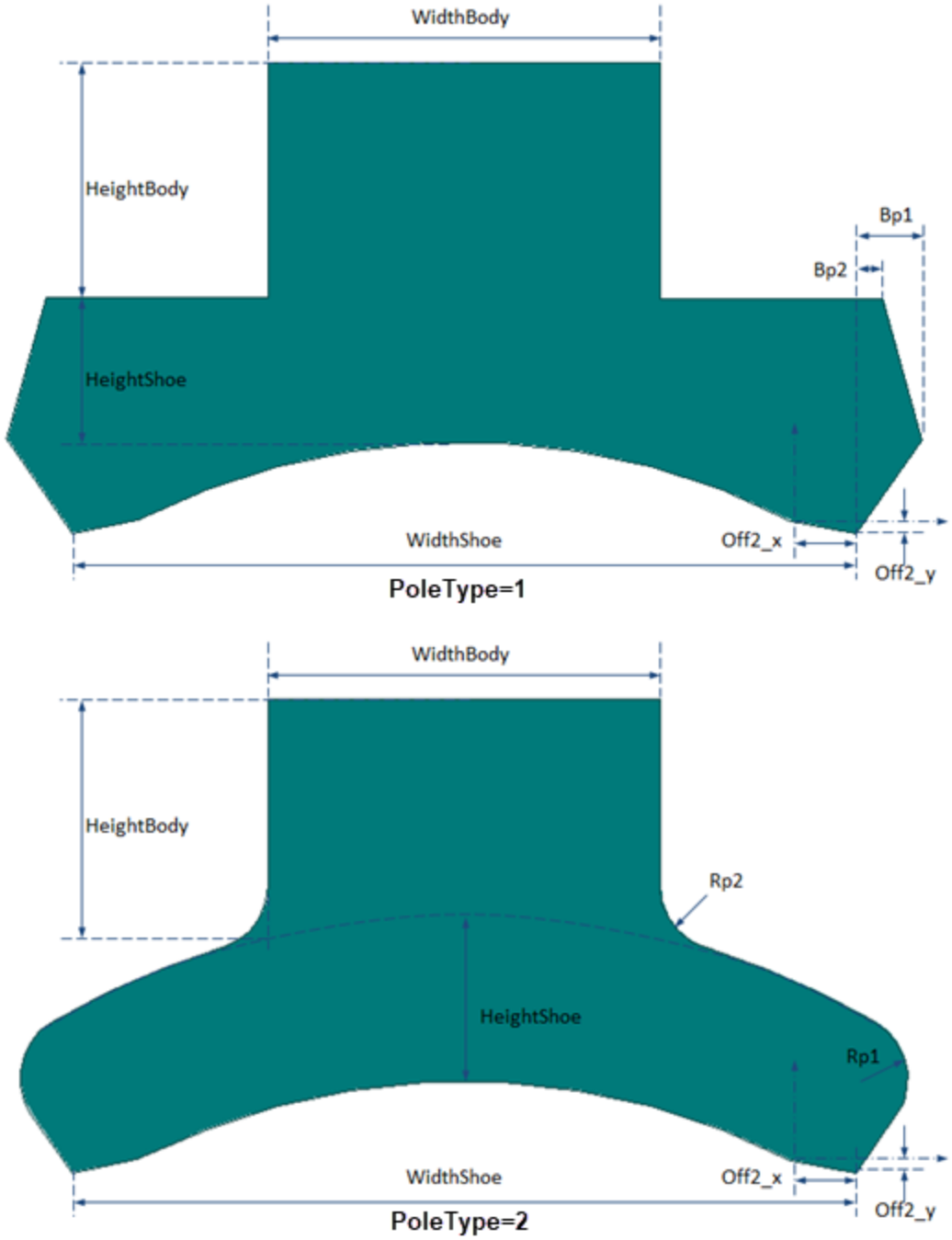


Figure 9-88

Two pole types available for outer cores

Figure 9-89 but supports only one pole type for inner cores.

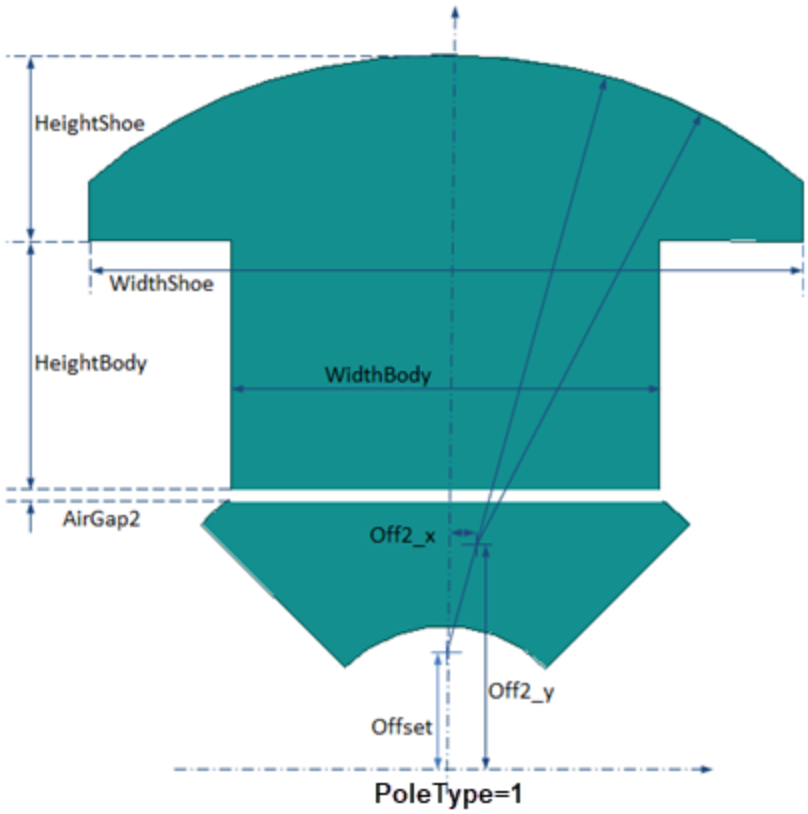


Figure 9-90

Only one pole type available for inner cores

Figure 9-91 Axial parameters for coils and dampers are shown in the following figures.

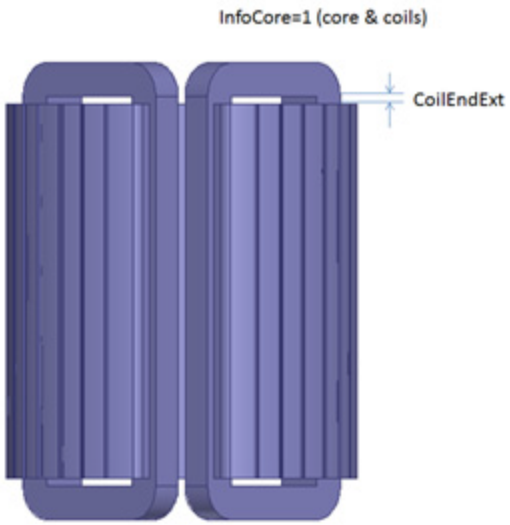


Figure 9-92 3D Salient Pole Core and coils (InfoCore set to 1)

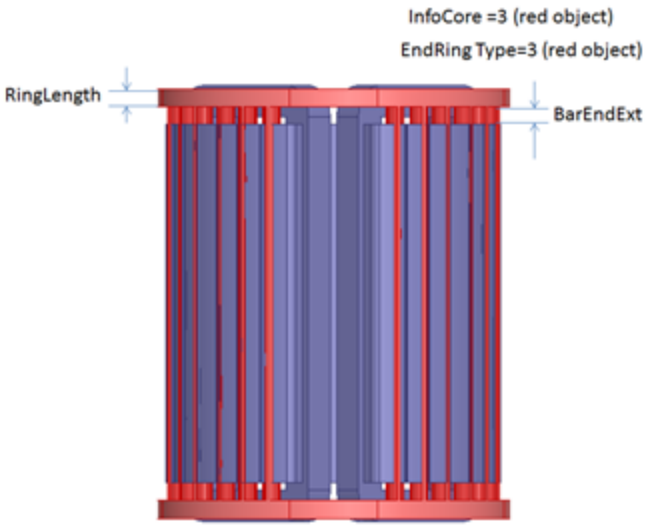


Figure 9-93 SalientPoleCore damper (InfoCore set to 3)

Creating an Inner or Outer Salient Pole Core

Set the value of **DiaYoke** and **DiaGap** as **DiaYoke**<**DiaGap** to create inner cores or **DiaGap**<**DiaYoke** for outer cores.

An example of an inner salient pole core is shown in Figure 9-86 . For an outer salient pole core, see Figure 9-94 .

You can set the values either when creating the core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Set the **InfoCore** value to 0.

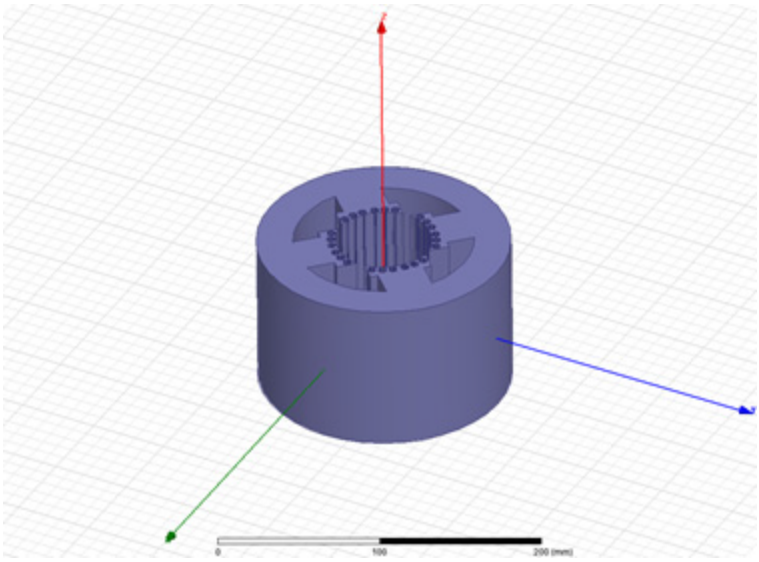


Figure 9-94 Outer salient pole core

Creating a Coil and its Terminals (Salient Pole)

You can create a coil of a salient pole core by manually setting the property of **InfoCoil** to 2. An example is shown below.

Its terminals are created manually by setting the property of **InfoCoil** to 4 or 5 (Terminal1 or Terminal2). The red sheet in the figure shows terminal 1 of the coil.

You can assign values for these properties either when creating a salient pole core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

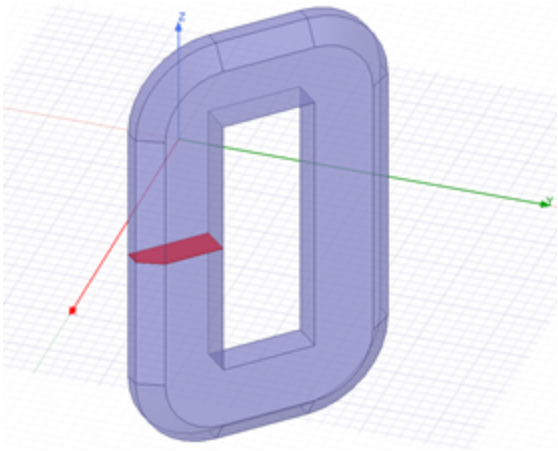


Figure 9-95 Salient pole core coil and its terminal

Creating a Damper (Salient Pole)

You can create a damper of a salient pole core by manually setting the property of **InfoCoil** to 3. An example is shown in Figure 9-93 as a red object.

Figure 9-96 ,Figure 9-97 , Figure 9-98 , and Figure 9-99 illustrate the shapes of the damper (InfoCore set to 3) with different EndRingType values. Gray objects are used as references.

You can assign the value of these properties either when creating a salient pole core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

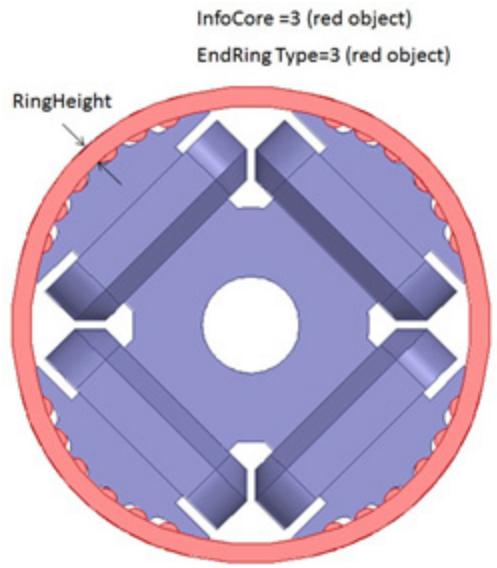


Figure 9-96 Damper with EndRing Type set to 3

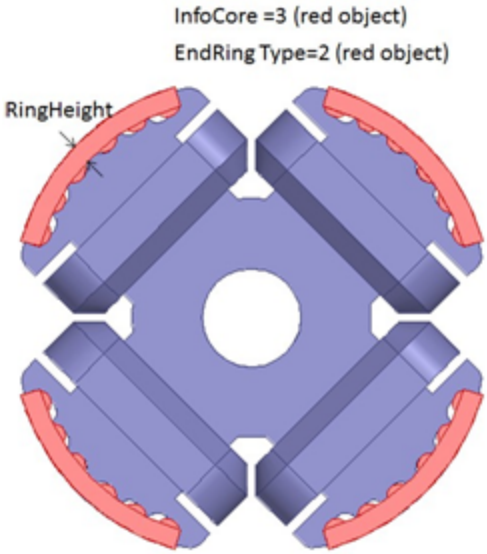


Figure 9-97 Damper with EndRing Type set to 2

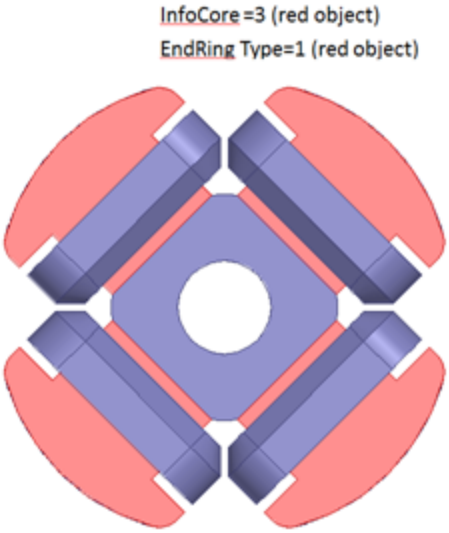


Figure 9-98 Damper with EndRing Type set to 1

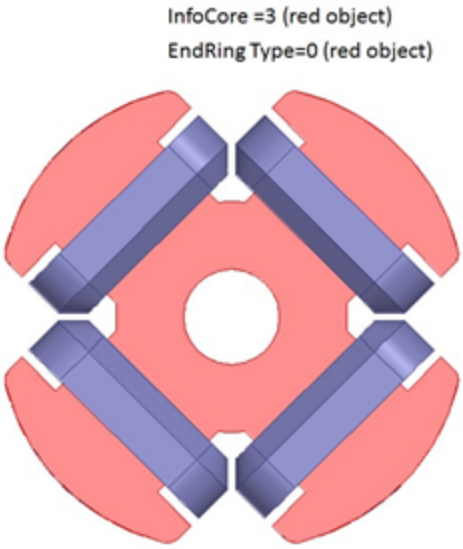


Figure 9-99 Damper with EndRing Type set to 0

Creating Poles or the Yoke

You can create poles or the yoke of SalientPoleCore UDP through assigning the **InfoCore** parameter to 6 or 7, respectively.

You can assign the value of this property either when creating a Salient Pole Core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Examples of poles and the yoke are shown in Figure 9-100 and Figure 9-101 , respectively.

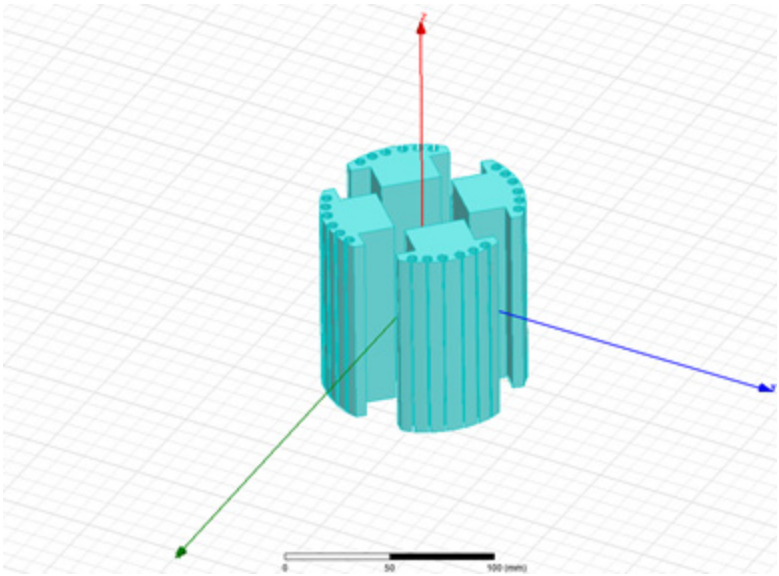


Figure 9-100 Poles of salient pole core

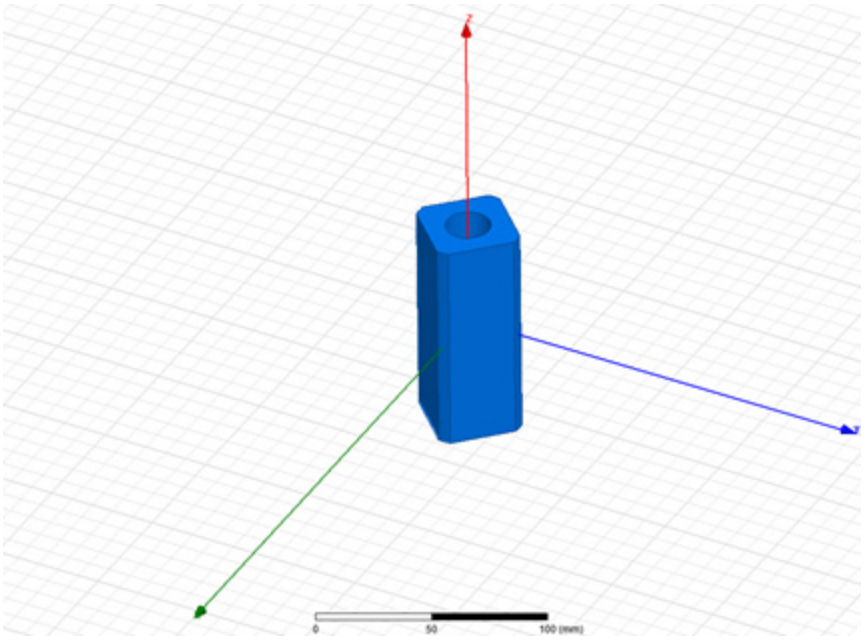
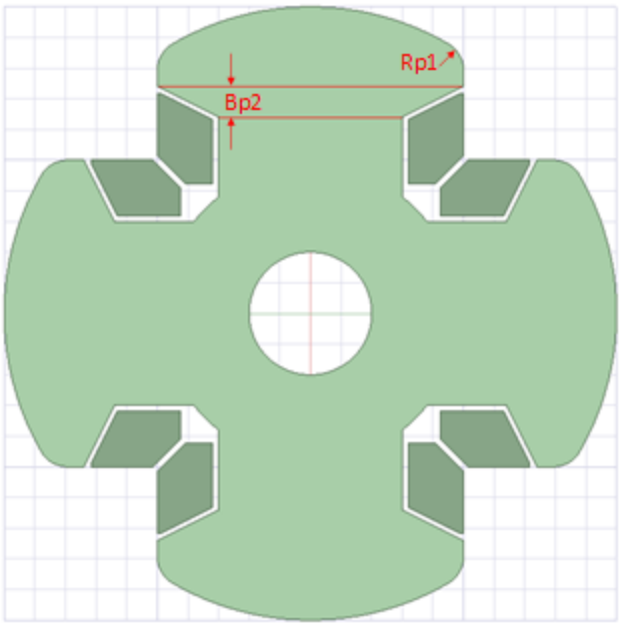


Figure 9-101 Yoke of salient pole core

Creating Inner Poles with Shoe Slope

For inner-core type-1 pole, parameters **Rp1/Bp1** and **Rp2/Bp2** stand for shoe-tip fillet radius and shoe-slope height, respectively, as shown below, where all coils are included.



Parameters **Rp1** and **Bp2** for inner core type 1 pole

DCMCore UDP

The DCMCore UDP is used to create a stator core with or without a compensating winding, a shunt and/or a series field winding, and a commutating pole with a commutating winding for DC machines. It can also create coil terminals for all windings for current assignment.

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores.
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores.
Length	Core length.
Skew	Skew angle in core length range.
FrameWidth	Overall width of a racetrack frame.
FrameThick	Frame thickness.
FrameLength	Frame length.
Poles	Number of poles.
PoleType	Pole type: 1 to 2.
Dmax	Diameter of shoe tip with maximum air gap length.
Bp0	Pole arc width with uniform air gap; 0 for eccentric air gap.
Bp1	Total pole width (the width between shoe tips).
Bp2/Rp0	Bp2 (max shoe width for pole type 1), or Rp0 (shoe fillet radius for pole type

Property	Description
	2).
Bp3/Rp1	Bp3 (min shoe width for pole type 1), or Rp1 (pole fillet radius for pole type 2).
Hp	Pole body height.
Bm	Pole body width.
FieldWndgs	Number of field windings, 2 for both series and shunt windings.
EndExt	Coil one-side end extended length.
SlotsPerPole	Compensating slots per pole distributed under pole arc surface.
Bc0	Opening width of compensating slots.
Hc0	Opening height of compensating slots.
Bc2	Width of compensating slots.
Hc2	Height of compensating slots.
CmpEndExt	One-side end extended length of compensating coils.
ComPoleWidth	Width of the commutating poles.
ComPoleHeight	Height of the commutating poles.
ComPoleLength	Length of the commutating poles.
ComShoeWidth	Shoe width of the commutating poles.
ComShoeHeight	Shoe height of the commutating poles.
ComShoeLength	Shoe length of the commutating poles.
ComGap2	Air gap length between commutating poles and the frame.
CmtEndExt	One-side end extended length of commutating coils.
LenRegion	Region length.
InfoCore	0: core & coils; 1: poles; 2: frame; 3: com poles; 4: shunt coil; 5: series coil; 6: com coil; 100: region.
Info Term	0: whole coil; 1: terminal1; 2: terminal2.

The DC Machine core is complex compared with other UDPs. When creating a DCM core in a Maxwell 3D design, it looks at first like the following figure in the coordinate system window.

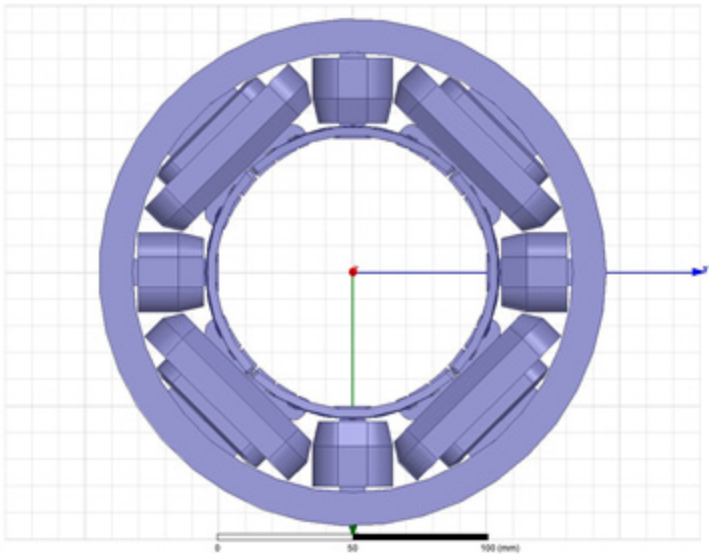


Figure 9-102 Initial DCM Core

In order to better describe the parameters in this UDP, a slot core was added inside the DCM core and their lengths were set to be 0. Also the **DiaGap** of slot core was changed to 92 to show the air gap between two UDPs. The model now looks like the following example:

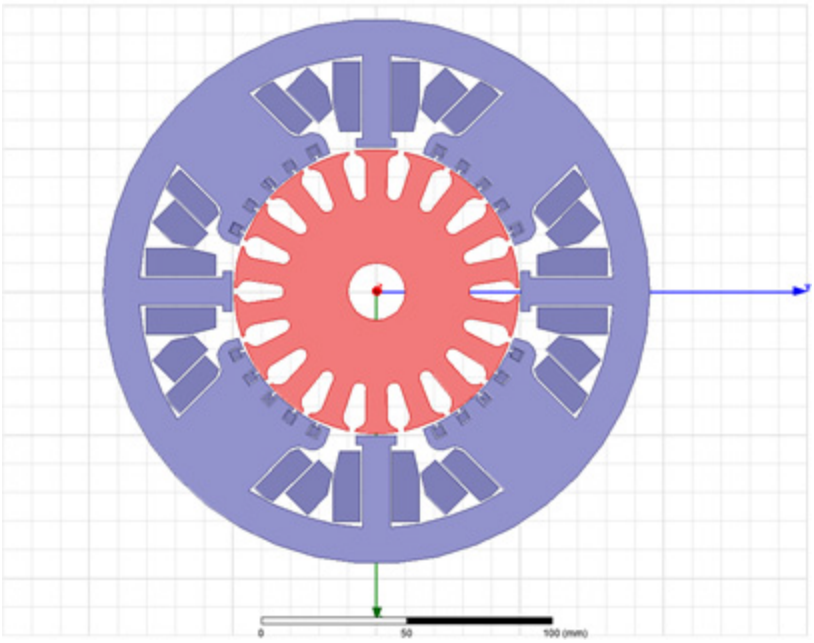


Figure 9-103 DCM Core Sheet (gray) with Slot Core Sheet (red)

The following figures show two different overviews of the gray, transparent DCM core and the referencing red slot core from different angles.

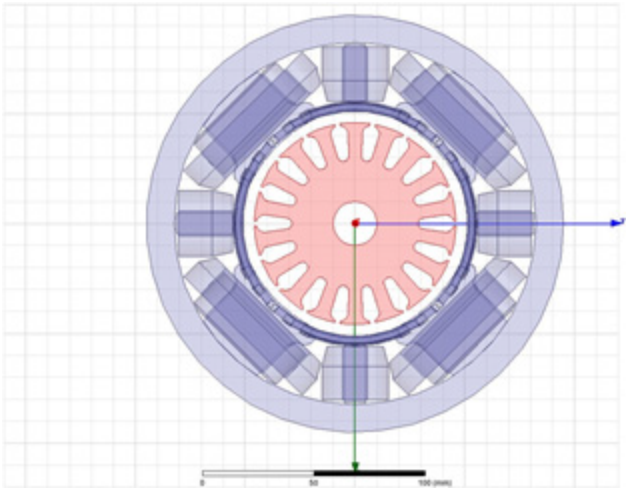


Figure 9-104 Transparent DCM core (gray) with Slot Core from different angles

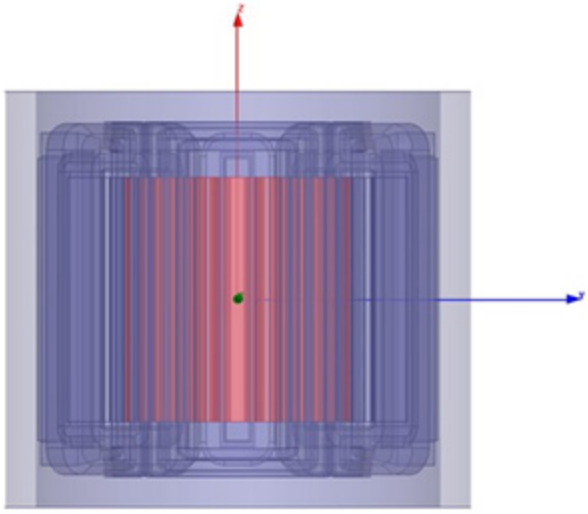


Figure 9-105 Transparent DCM core (gray) with Slot Core from different angles

The following figures show the parameters of a DCM core.

Note	1. A slot core in red is drawn as reference. 2. Because the structure of a DCM core is complicated and there are many parameters, figures are drawn using different **InfoCore** values of DCM core UDP to clearly describe parameters. 3. Figure 9-106 and Figure 9-107 show parameters of two different pole types in the DCM core UDP. In these two figures, **Bp0** is not 0, whereas in Figure 9-108 **Bp0**

	is 0, which means the air gap is eccentric. 4. **Dmax**, by definition, is the diameter of shoe tip with maximum air gap length, which means **Dmax**=Drout+2*Gmax. “Drout” means the outer diameter of the rotor (which means **DiaGap** of slot core in this example and is shown in Figure 9-108), and “Gmax” is the maximum air gap between poles and rotor, which is shown in Figure 9-106 and Figure 9-107 . 5. Correspondingly, there is **Dmin** (which is called **DiaGap** in DCM UDP and Figure 9-108) and **Gmin**. Their relationship is **Dmin**=Drout+2*Gmin. They are also shown inFigure 9-106 , Figure 9-107 , and Figure 9-108 . 6. The value of **ComGap2** in Figure 9-108 is 0, whereas it is positive in Figure 9-109 . In order to make the value of **ComGap2** larger than 0, make sure to reserve enough space for **ComGap2**. This can be done by reducing the value of **ComPoleHeight** or **ComShoeHeight** or both.

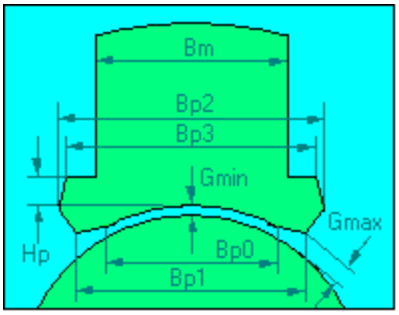


Figure 9-106 Parameters of PoleType 1

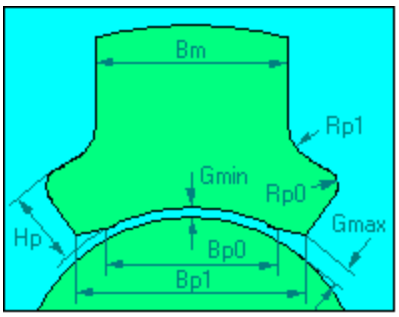


Figure 9-107 Parameters of PoleType 2

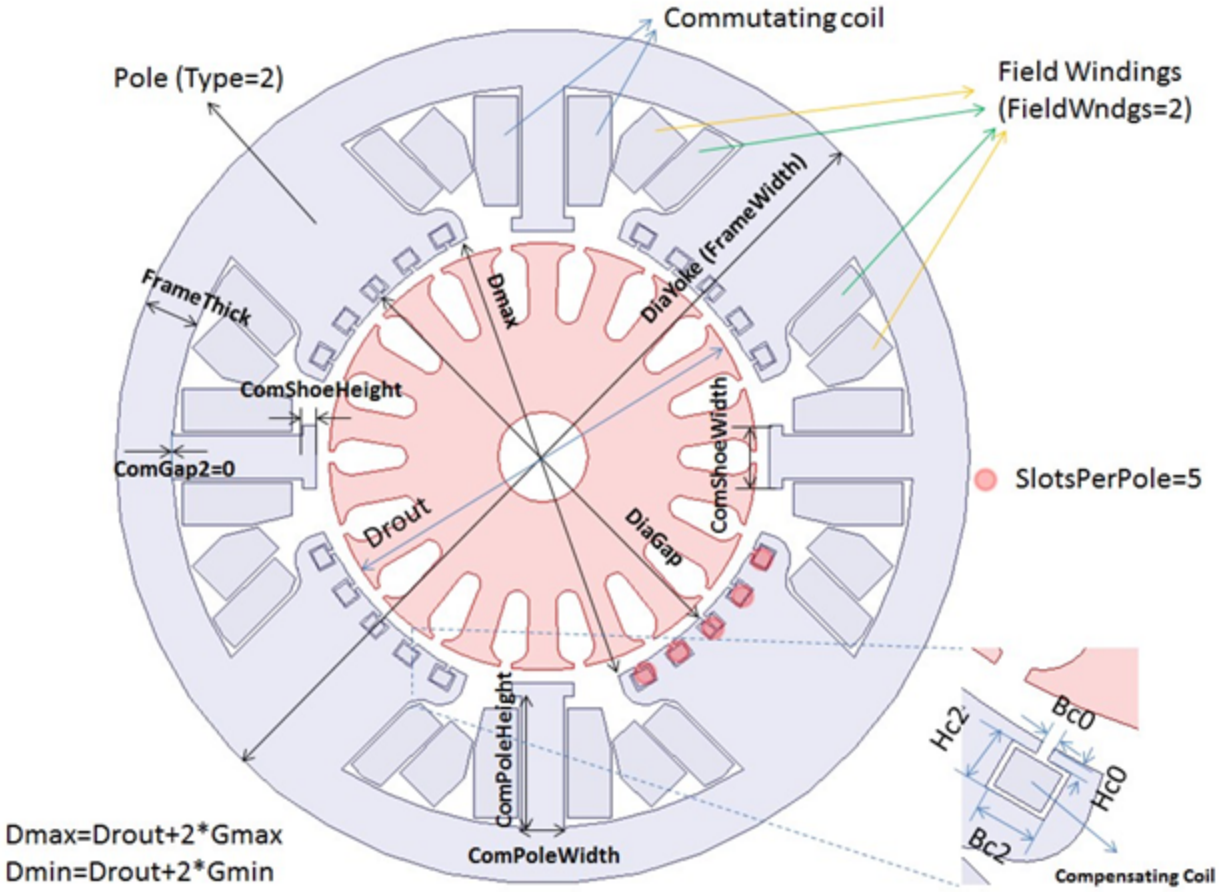


Figure 9-108 Parameters of DCM Core on 2D plane

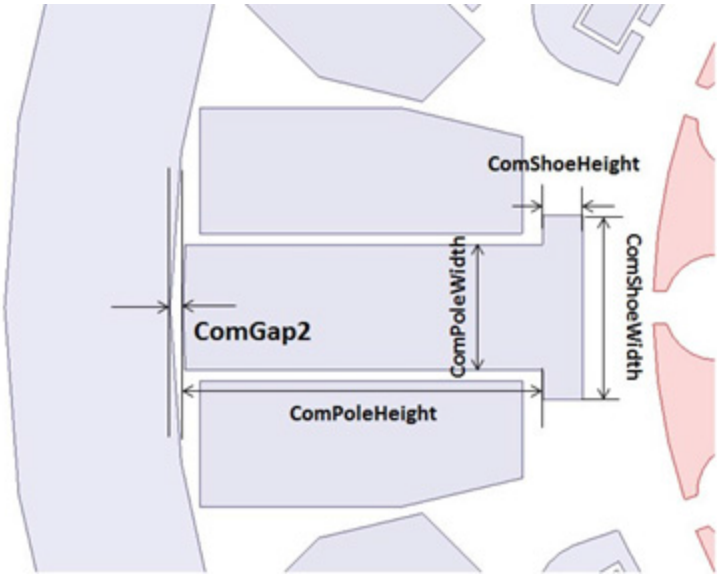


Figure 9-109 Parameters of a commutating pole (when ComGap is non-zero)

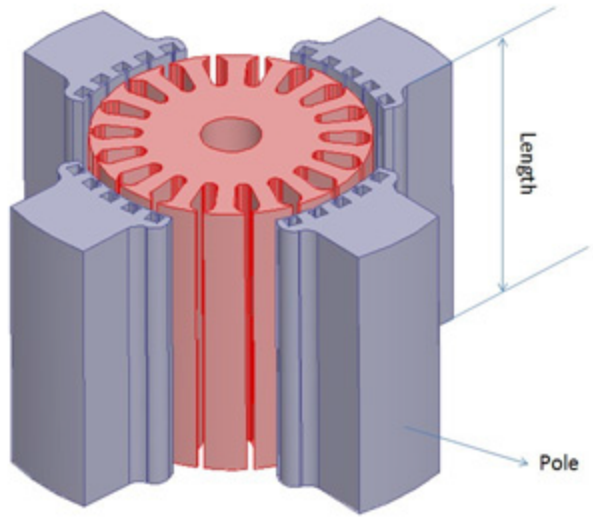


Figure 9-110 Length of poles

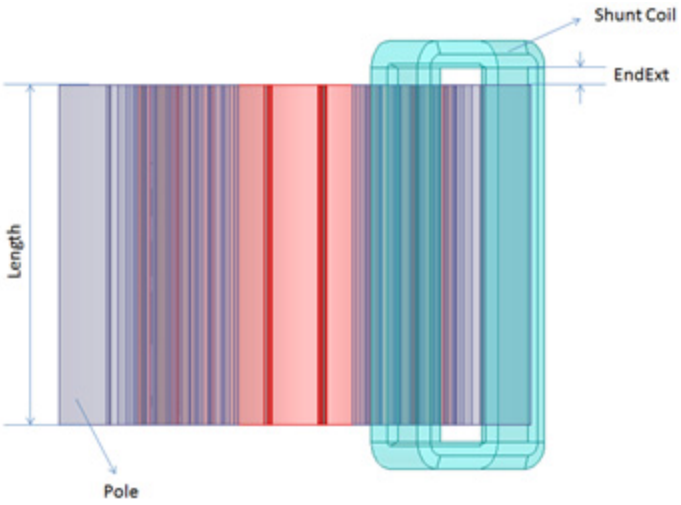


Figure 9-111 Axial parameters of poles and field windings

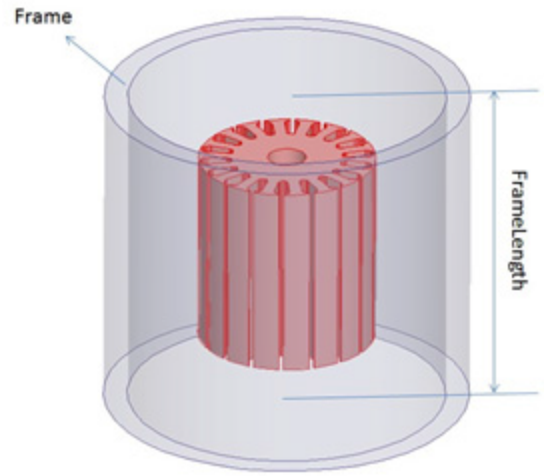


Figure 9-112 Length of frame

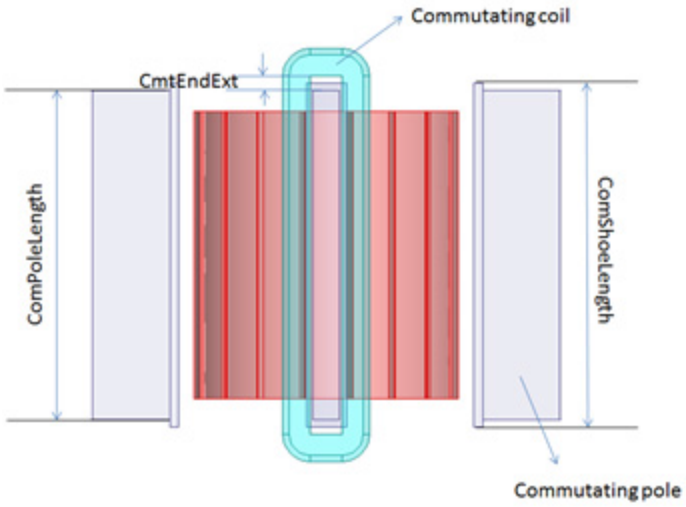


Figure 9-113 Axial parameters of commutating poles and commutating coils

Creating an Outer DC Machine Core and All Coils

This core is used only as outer core, which means the value of **DiaYoke** is larger than that of **DiaGap**. Also,set both **InfoCore** and **Info Term** values to be 0. You can set the value either when creating the core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

An example of an outer DC machine core is shown in the following figure:

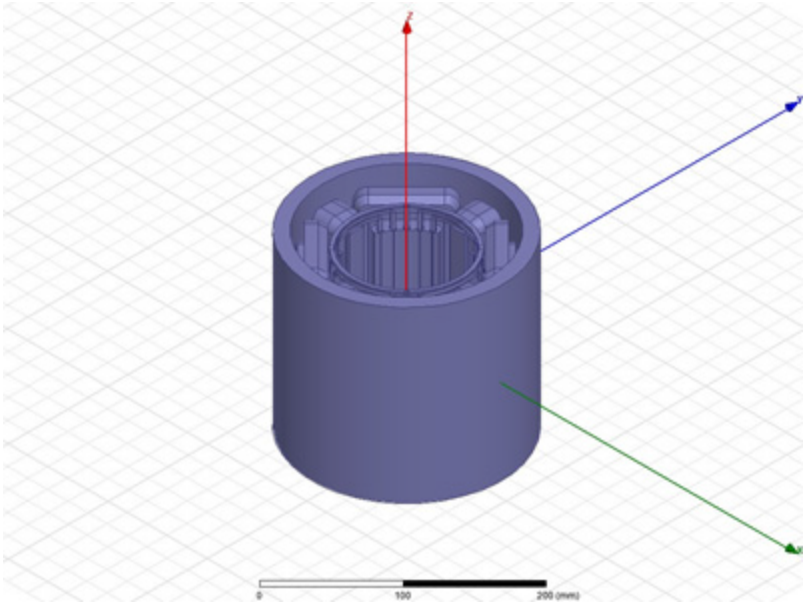


Figure 9-114 Outer DC Machine core and all coils

Creating Main Poles, Frame and Commutating Poles Separately

You can create the main poles of a DC Machine core by manually setting the property of **InfoCore** to 1 and **Info Term** to 0. The result is shown as the gray object in Figure 9-110 and Figure 9-111 .

You can create the frame of a DC Machine core by manually setting the property of **InfoCore** to 2 and **Info Term** to 0. The result is shown as the gray object in Figure 9-112 .

You can create the commutating poles of a DC Machine core by manually setting the property of **InfoCore** to 3 and **Info Term** to 0. The result is shown as the gray object in Figure 9-113 .

You can assign the values of these properties either when creating a DC Machine core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Creating a Shunt Coil and its Terminals

You can create a shunt coil of a DC Machine core by manually setting the property of **InfoCore** to 4 and **Info Term** to 0. The result is shown as a blue object in Figure 9-111 .

You can also create its terminals by setting the property of **InfoCore** to 4 and **Info Term** to 1 or 2 (for terminal1 and terminal2, respectively).

You can assign the values of these properties either when creating a DC Machine core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Creating a Series Coil and its Terminals

You can create a series coil of a DC Machine core by manually setting the property of **InfoCore** to 5 and **Info Term** to 0. An example is shown in the following figure.

You can also create its terminals by setting the property of **InfoCore** to 5 and **Info Term** to 1 or 2 (for terminal1 and terminal2, respectively).

You can assign the values of these properties either when creating a DC Machine core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

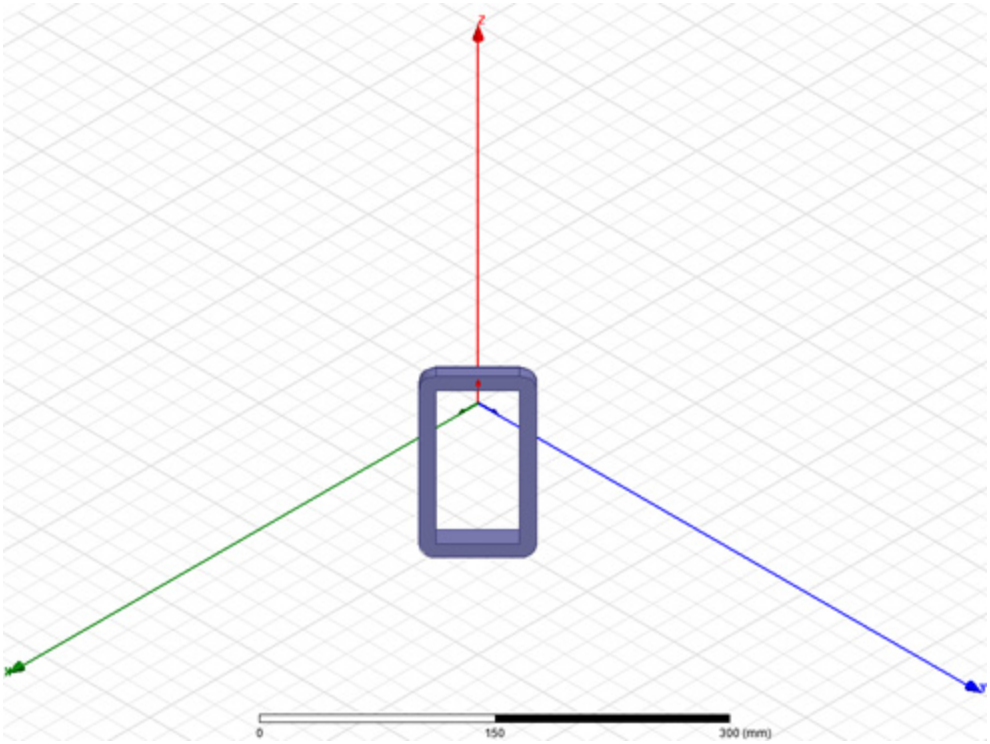


Figure 9-115 A series coil of DCM Core

Creating a Commutating Coil and its Terminals

You can create a commutating coil of a DC Machine core by manually setting the property of **InfoCore** to 6 and **Info Term** to 0. The result is shown as a blue object in Figure 9-113 .

You can also create its terminals by setting the property of **InfoCore** to 6 and **Info Term** to 1 or 2 (for terminal1 and terminal2, respectively).

You can assign the values of these properties either when creating a DC Machine core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

ShadedPoleCore2 UDP

The ShadedPoleCore2 UDP is used to create the stator core for shaded-pole induction motors.

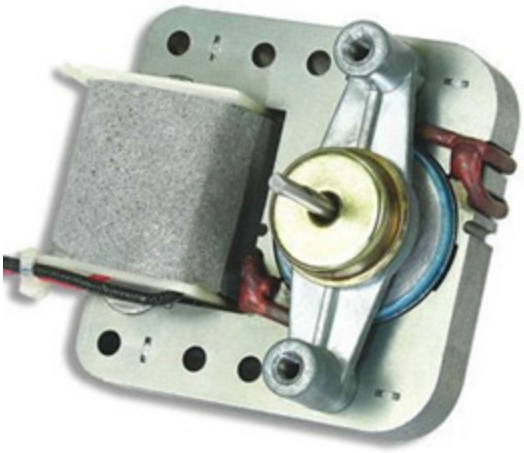


Figure 9-116 Shaded pole induction motor example 1

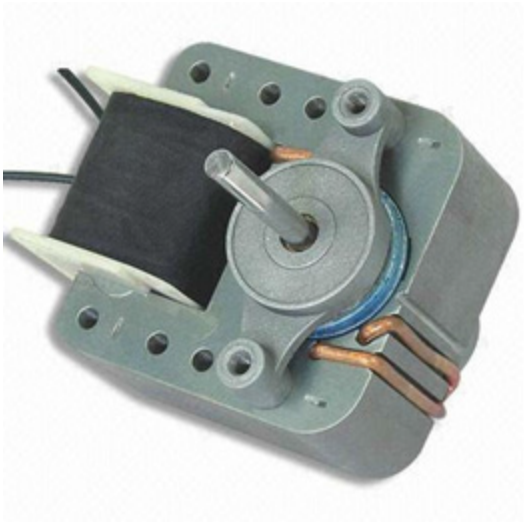


Figure 9-117 Shaded pole induction motor example 2

Property	Description
DiaGap	Core diameter on gap side, or inner diameter.
HighCore	Core height in d-axis direction.

Property	Description
WidthCore	Core width in q-axis direction.
WidthPole	WidthPole = DiaGap + 2 * (pole bridge thickness).
Length	Core length in the axial direction.
WidthYoke1	Width of the winding-core yoke.
WidthYoke2	Width of the parallel yokes.
Slots	Number of slots per pole (1 or 2).
Hs0	Slot opening height.
Bs0	Slot opening width.
Bs1	Slot diameter.
DiaCond	Diameter of the short-circuit solid conductor.
AngleOff	Angle offset in mechanical degrees from the first slot to d-axis.
SlotPitch	Slot pitch in mechanical degrees for two slots.
Dist1	Outer-slot center-to-center distance when Slots = 2.
DiaS2	Additional outer-slot diameter.
Dist2	Additional outer-slot distance.
InfoEnd	0: Separated end connectors; 1: Jointed end connectors.
GapDelta	Maximum gap - minimum gap.
AngleBegin	Begin angle of maximum gap in mechanical degrees.
AngleEnd	End angle of maximum gap in mechanical degrees.
RadFillet	Radius of outer-corner fillet.
RadDelta	Step depth in outer-corner fillet.
DiaHole	Diameter of holes for shaft bearing covers.
SpanHole	Hole span in d-axis direction.
LenRegion	Region length.
InfoCore	0: Core; 1: All; 2: Coil; 3: Cond1; 4: Cond2; 5: Hole filler; 100: Region.
InfoTerm	0: Whole coil; 1: Terminal1; 2: Terminal2.

These parameters are used in the following figures.

Note	• The number of parameters is large in this UDP, so different colors are used to indicate different parameters in the stator bore (area of the circle of **DiaGap**). Each parameter is shown with an arc/line and text in the same color. • For parameters such as **Length** and parameters related to slots, refer to SlotCore UDP parameters.

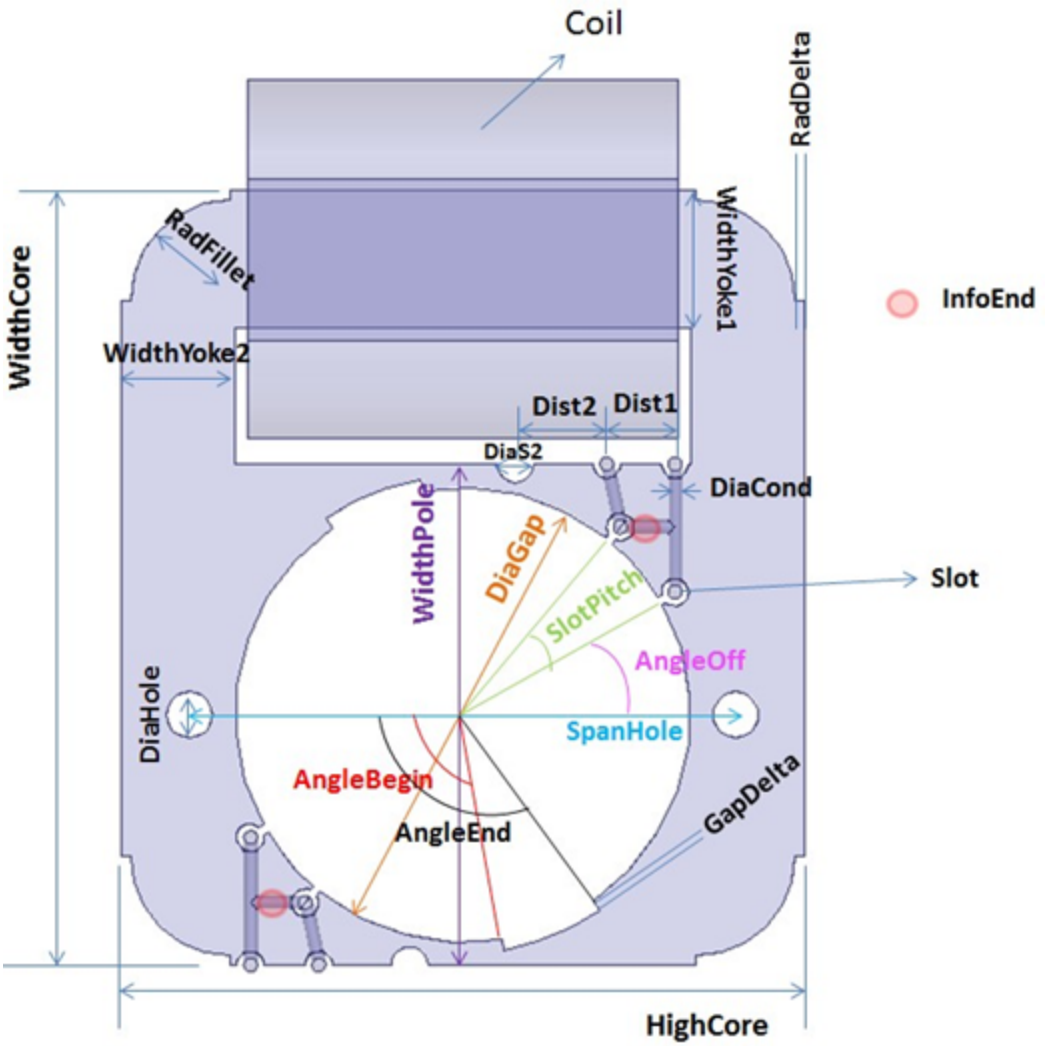


Figure 9-118 Parameters of ShadedPoleCore2 (InfoCore set to 1, Info Term set to 0)

Creating an Outer Shaded Pole Core

This core is used only as an outer core. There is no **DiaYoke** parameter in this UDP, so to create an outer shaded pole core,set both **InfoCore** and **Info Term** values to be 0. You can set the values either when creating the core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

An example of the core for a ShadedPoleCore2 is shown in the following figure.

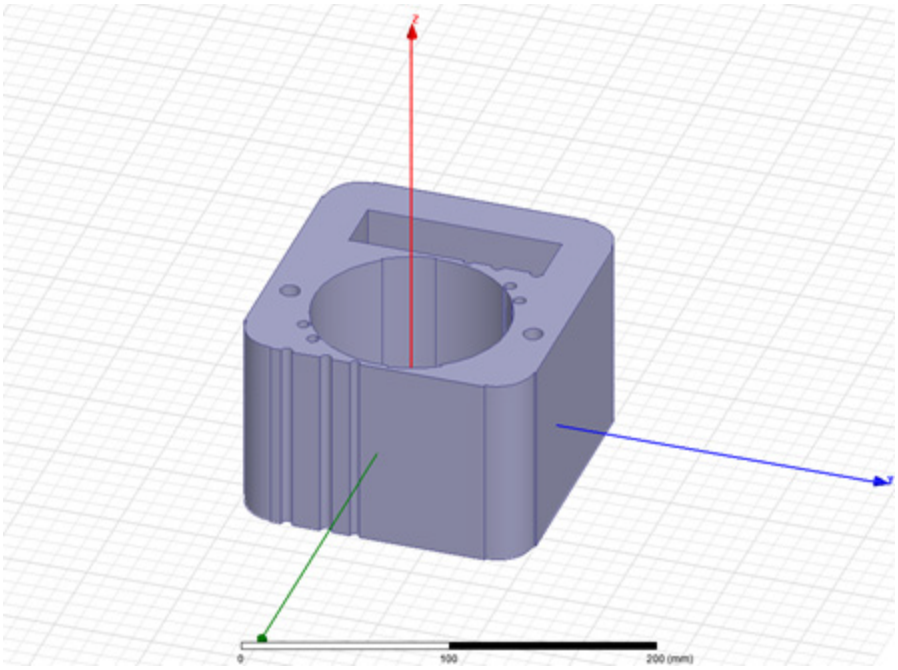


Figure 9-119 Core of ShadedPoleCore2

Creating a Coil and its Terminals (Shaded Pole Core)

A coil of a shaded pole core is created by manually setting the property of **InfoCore** to 2 and **Info Term** to 0. An example of the coil for a ShadedPoleCore2 UDP is shown in Figure 9-120 .

You can create its terminals by setting the property of **InfoCore** to 2 and **Info Term** to 1 or 2 (for terminal1 and terminal2 respectively). Figure 9-121 gives an example of a terminal.

You can assign the values of these properties either when creating a shaded-pole core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

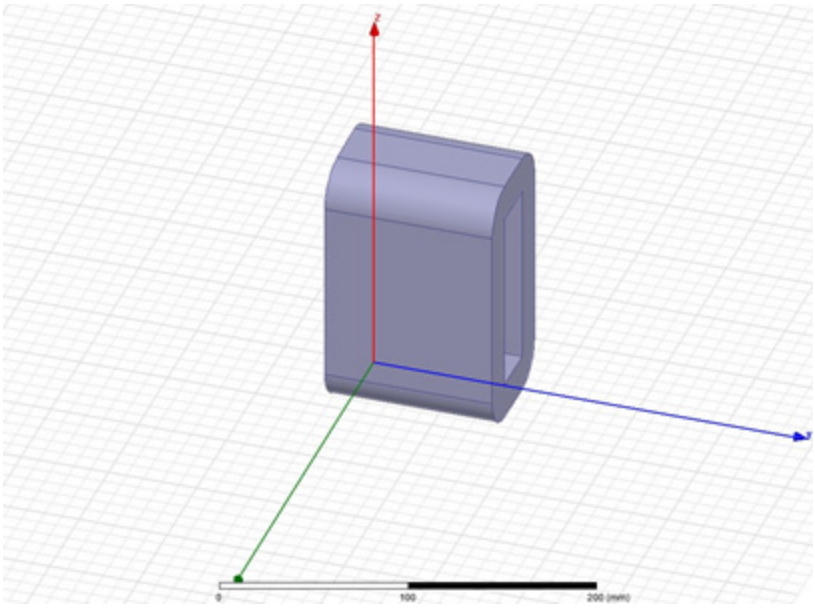


Figure 9-120 A coil of a ShadedPole Core

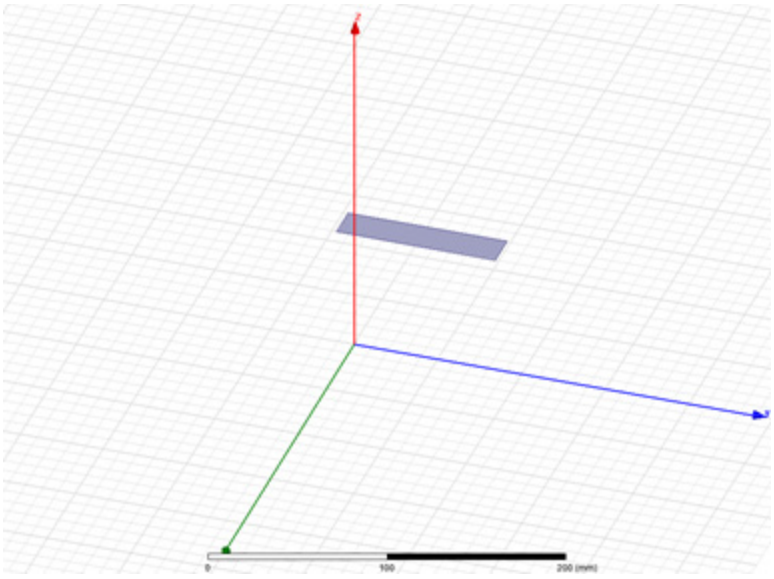


Figure 9-121 A terminal of a coil for a ShadedPole Core

Creating a Jointed or a Separated Shaded Conductor and Its Terminals

You can create a jointed conductor by manually setting the parameter **InfoCore** to 3 and **Info Term** to 0. At the same time, set **InfoEnd** to 1 to make the conductor connected. An example of a jointed conductor is shown in the following figure.

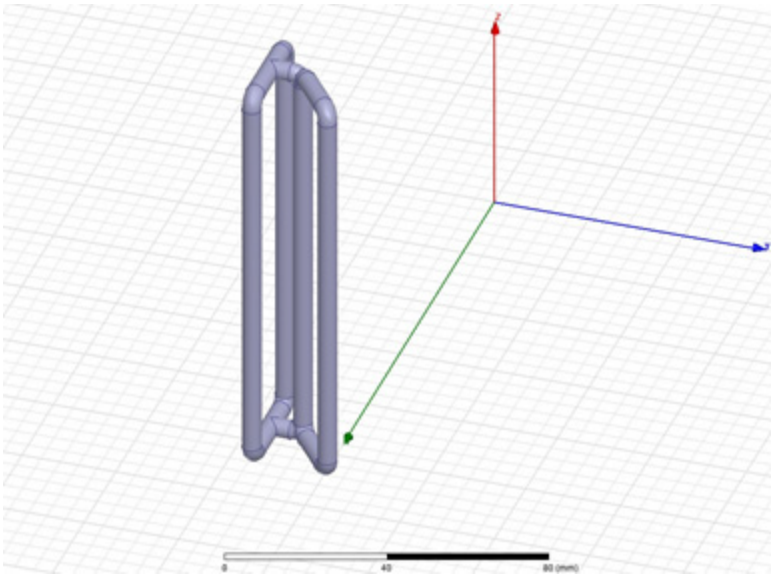


Figure 9-122 A jointed shaded conductor for a ShadedPole Core

You create a separated conductor by changing **InfoEnd** to 0. Set **InfoCore** to 3 or 4 to create conductor 1 or 2, respectively. **Info Term** remains 0. The following figure provides an example of the separated conductor.

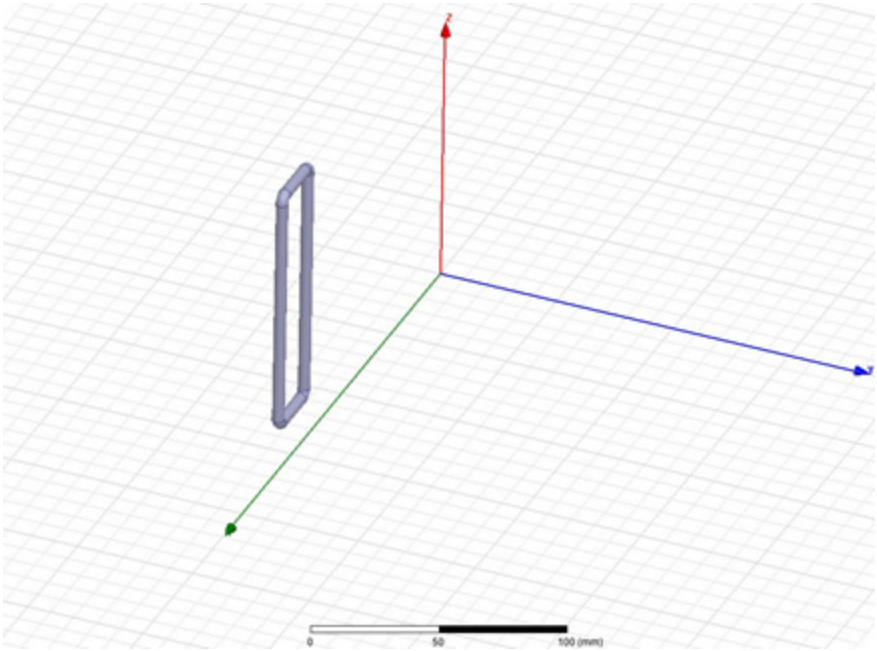


Figure 9-123 A separated shaded conductor for a ShadedPole Core

You create terminals of a conductor by changing the **Info Term** to 1 or 2 for terminal1 or terminal2, respectively. At the same time, **InfoCore** should be 3 or 4. Figure 9-124 shows the results of setting the parameters.

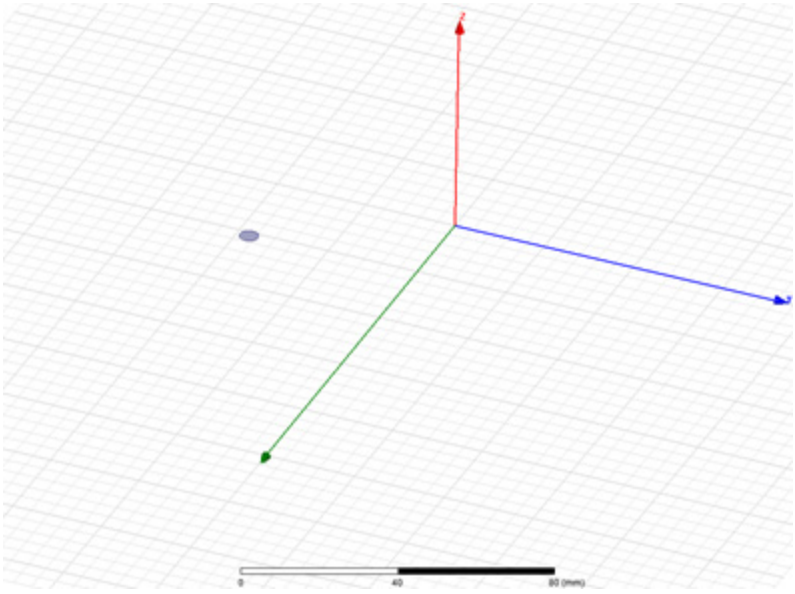


Figure 9-124 A terminal of conductor for a ShadedPole Core

Creating a Core with Poly Holes

Normally a Shaded Pole Core UDP has only one pair of holes. In order to create a poly-hole Pole Core UDP, you need to use subtraction and move commands. Detailed steps are as follows.

1. Create a core as described in Creating an Outer Shaded-Pole Core.

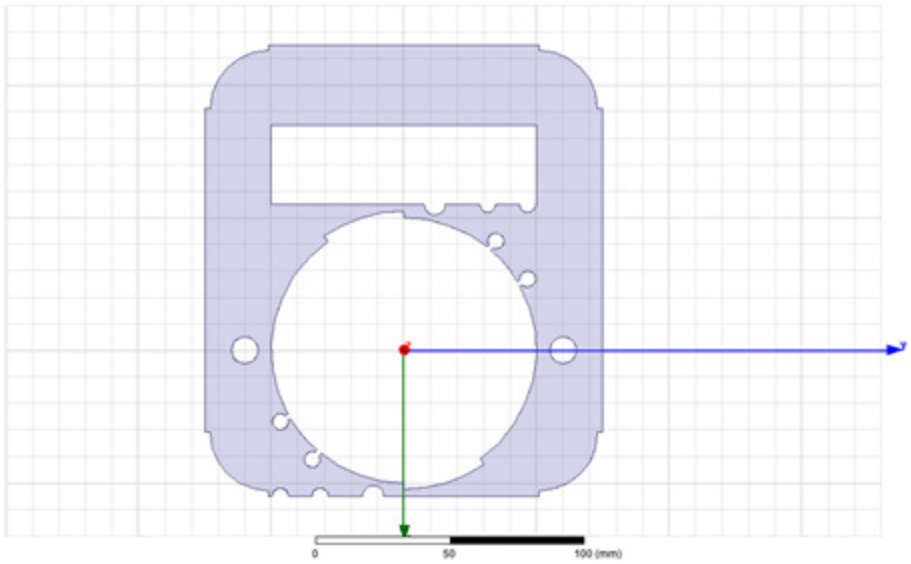


Figure 9-125 Original ShadedPoleCore

2. Make a copy of the current core.
3. Select the desired inner core in the history tree, and right-click to choose **Edit>Copy**.

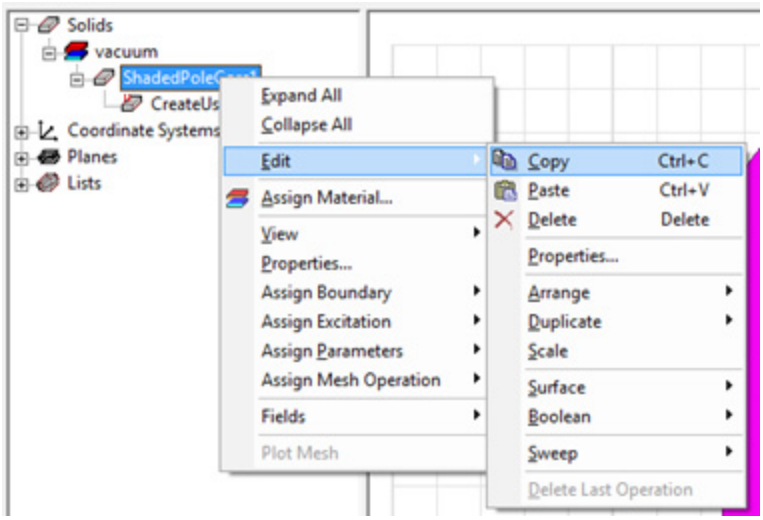


Figure 9-126 Copy ShadedPoleCore

4. Then right-click the same object in the history tree to choose **Edit>Paste**. This process is shown in the following figure:

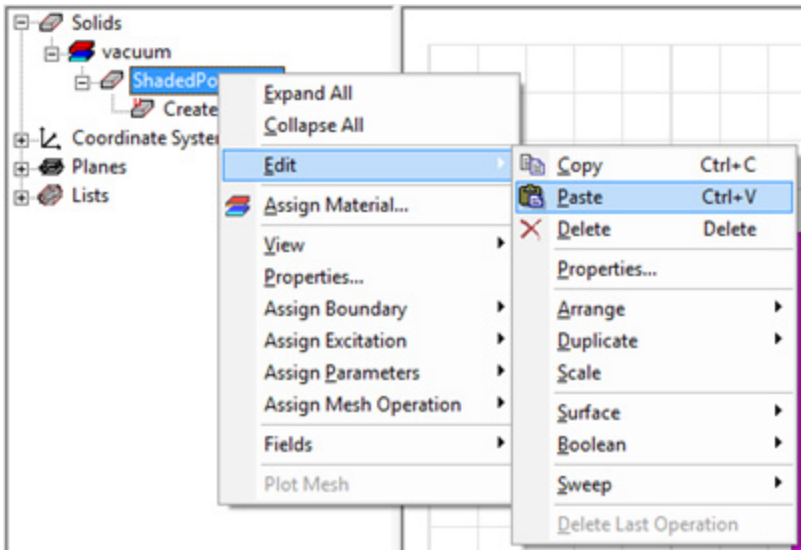


Figure 9-127 Paste ShadedPoleCore

5. Change the new object to a Hole filler.

   Click on **CreateUserDefinedPart** of ShadedPoleCore2 objectand set **InfoCore** property to **5** in the pop-up window. The objects in the coordinate system window are shown in the following figure:

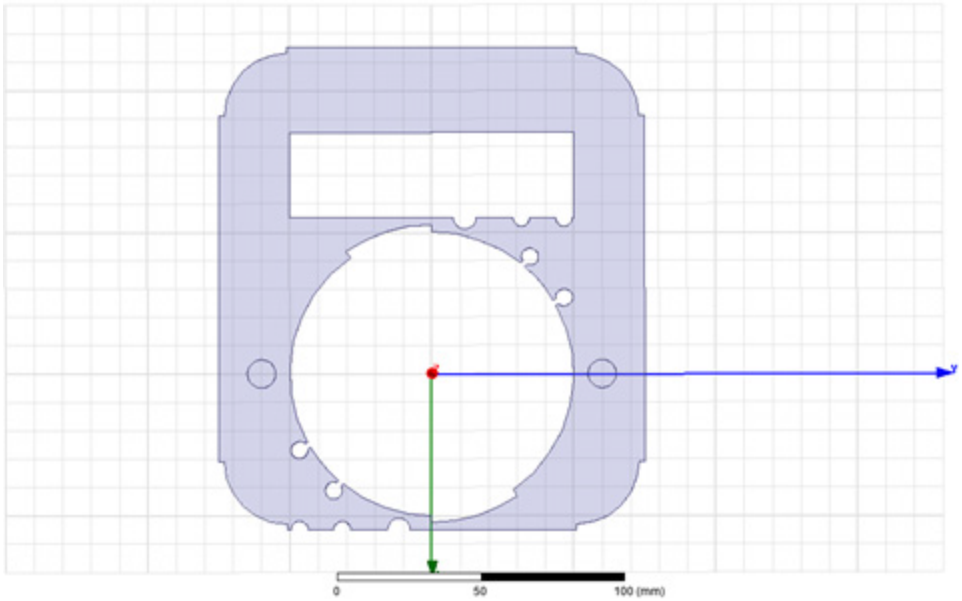


Figure 9-128 Change the InfoCore of the second ShadedPoleCore

6. Move the hole filler to another position for subtraction.
7. Select the hole filler (ShadedPoleCore2) in the history tree.
8. Click **Edit**>**Arrange**>**Move** from the menu. The modeler switches to move mode and prompts you for a reference point in the status bar.

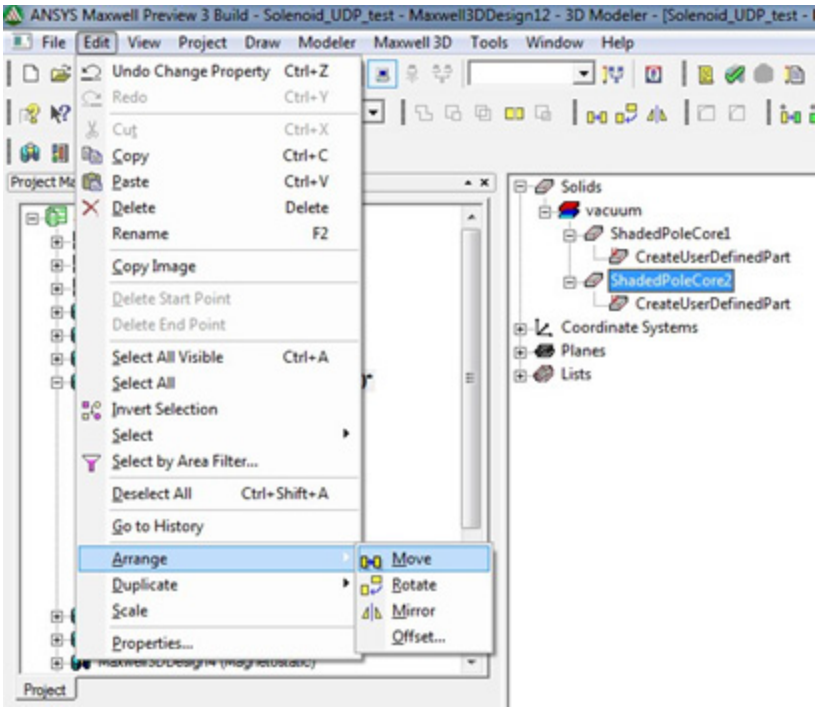


Figure 9-129 Move the position of the hole filler

9. Enter a reference point by clicking at the origin. Another option is to enter **(0, 0, 0)** in the keyboard entry area of the status bar, and press **Enter**.
10. Enter a target point along the Z-axis by clicking the mouse along the axis. Another option is to enter **(30, 0, 0)** in the keyboard entry area, and press **Enter**, which is shown in the

following figure.

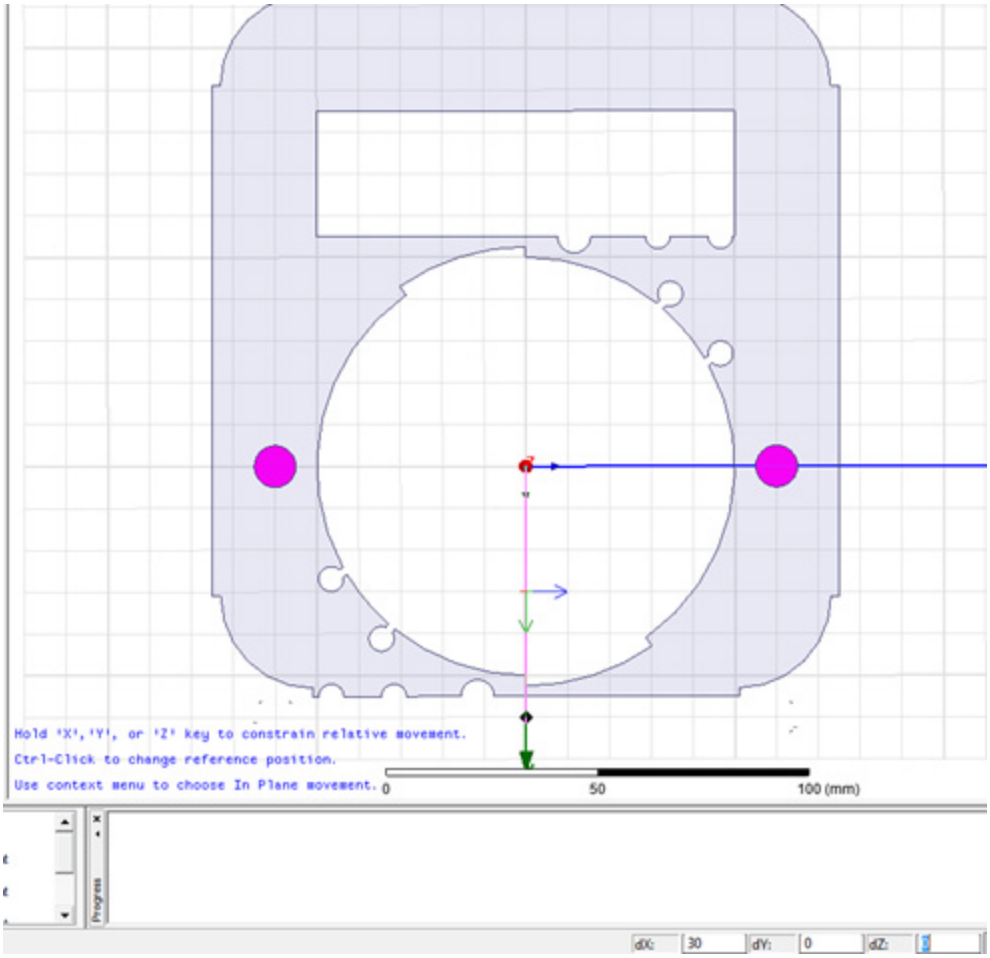


Figure 9-130 Enter a target position

11. Choose the core (ShadedPoleCore1) and the hole filler (ShadedPoleCore2) to make a subtraction.
12. Select both the core and hole filler by pressing **Ctrl** on the keyboard.

13. Click **Modeler**>**Boolean**>**Subtract** to subtract the two selected objects.

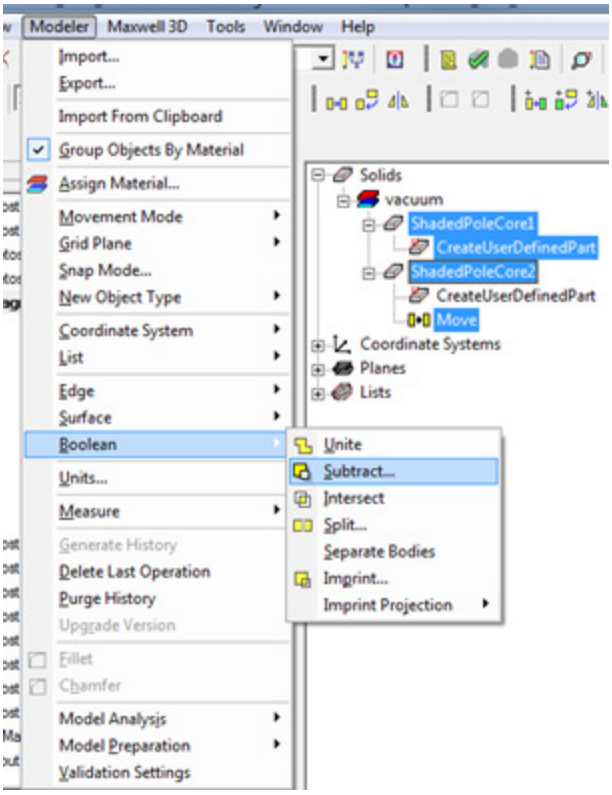


Figure 9-131 Choose Subtract operation from Boolean operation

14. In the **Subtract** window, put core (ShadedPoleCore1) in the **Blank Parts** and hole filler (ShadedPoleCore2) in **Tool Parts**.

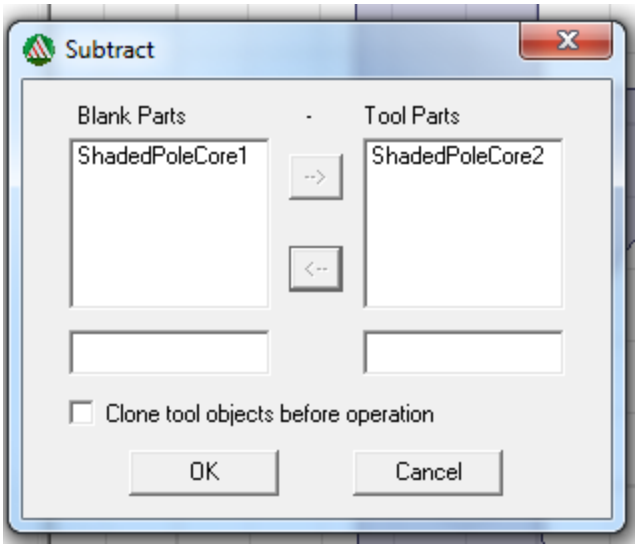


Figure 9-132 Select the Blank Parts and Tool Parts

15. Click **OK** to finish subtraction. The core look likes the following figure:

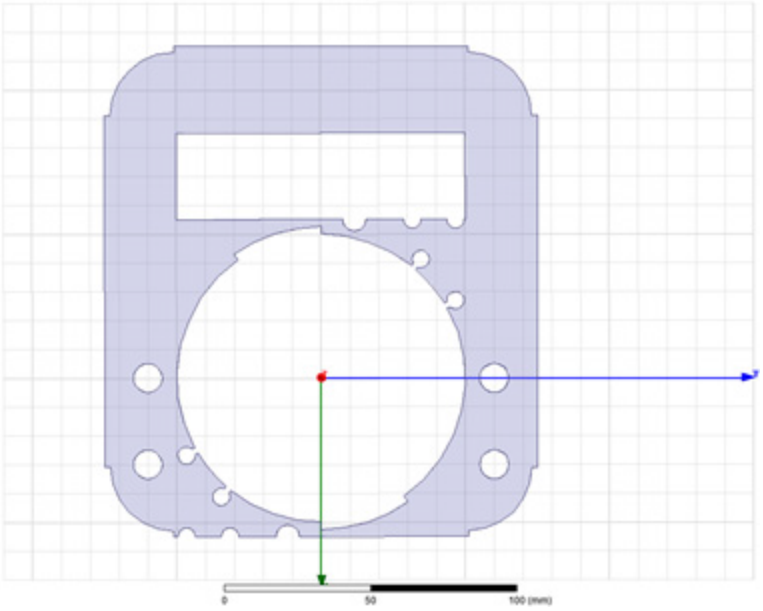


Figure 9-133 Finish subtraction

Another option is to copy and paste multiple cores, change them to hole fillers and move them to different positions to create a core with more than 2 pairs of holes. The following figure shows a sample hole:

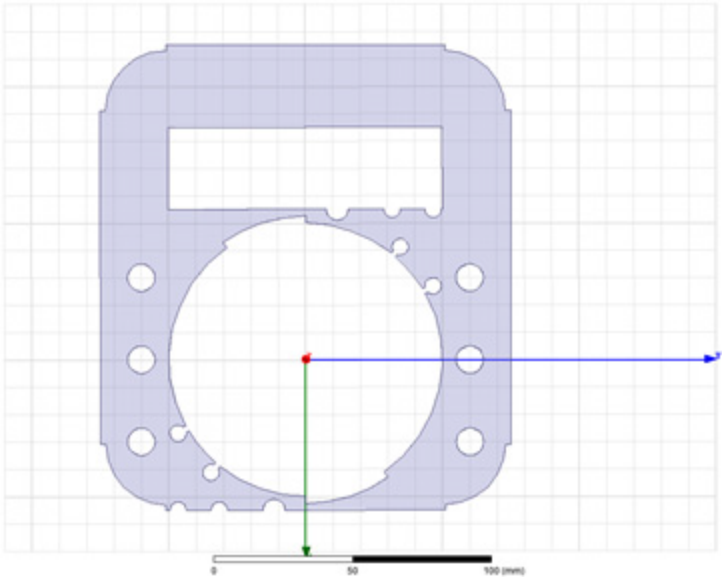


Figure 9-134 ShadedPoleCore with multiple poles

UnivMCore UDP

The UnivMCore UDP is used to create the stator core and the field winding of a universal motor. It can also create a coil terminal for current assignment.

Property	Description
DiaGap	Core diameter on gap side, or inner diameter.
DiaYoke	Core diameter on yoke side, or outer diameter.
Length	Core length.
Skew	Skew angle in core length range.
Poles	Number of poles.
PoleType	Pole type: 1 to 2.
WidthCore	Overall width of a racetrack core.
Embrace	Pole embrace (the ratio of pole arc to pole pitch).
Offset	Pole arc offset.
WidthPole	Minimum width at pole root.
Ty	Yoke thickness.
Ts	Shoe-tip thickness.
R1	Radius of the screw holes in pole center.
R2	Radius of the side fillet arc at pole root.
R3	Radius of the side fillet arc center layout circle.
R4	Radius of the shoe connecting arc, 0 for auto-design.
R5	Inner radius of the screw holes in between two poles.
R6	Outer radius of the screw holes in between two poles.
EndExt	Coil one-side end extended length.
LenRegion	Region length.
InfoCore	0: core; 1: coils; 2: one coil; 3: terminal1; 4: terminal2; 100: region.

These parameters are used in the following figures:

Note	• In figures with both gray and red objects, gray objects are used as references. • Universal motor core is used as outer core, which means the value of DiaYoke is larger than that of DiaGap. • Although R4 is drawn in Figure 9-135 , it is not easy to be understood. It is the radius of a circle which is externally tangential to the circle of radius R2 and internally tangential to the circle of diameter of Ts on the same side of a pole. R4

	and the two circles that it is tangential to are marked in red. When the value of R4 is 0, the length of it is automatically designed.

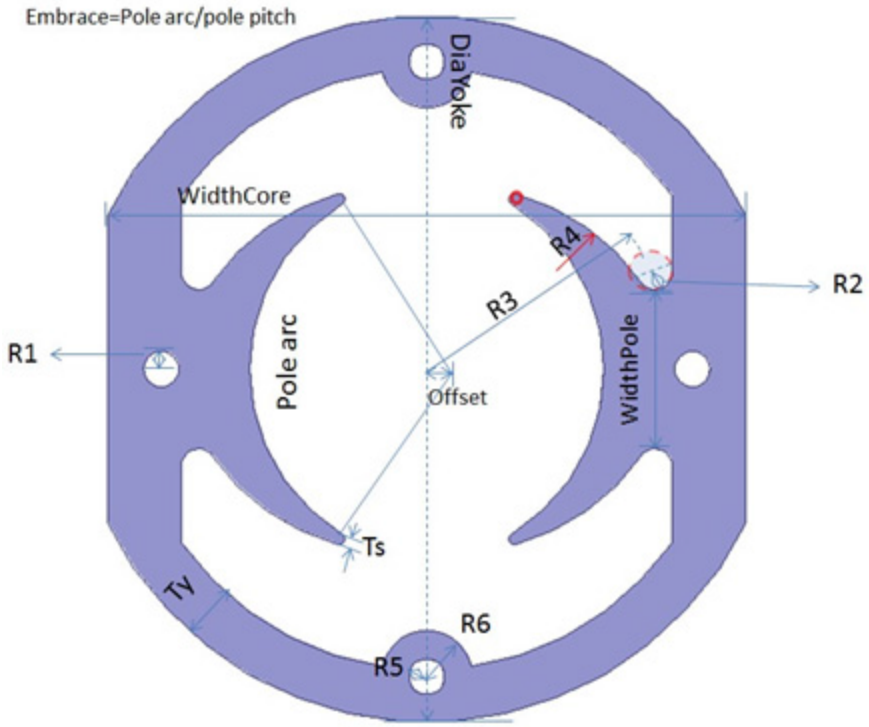


Figure 9-135 Parameters in UnivM Core

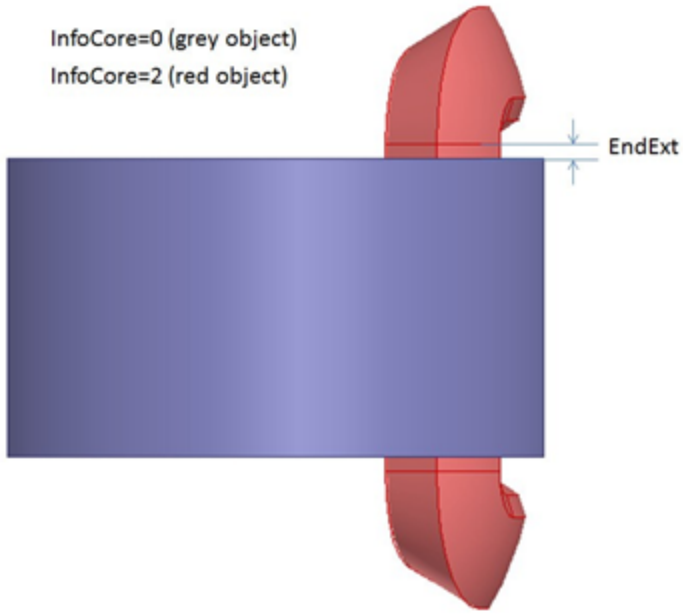


Figure 9-136 Two UnivM Core with different InfoCore value

Creating an Outer Universal Motor Core

This core is used only as an outer core. You can set the value of **DiaYoke** and **DiaGap** as **DiaYoke**>**DiaGap** to create an outer universal motor core. You can set the values either when creating the core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Set the **InfoCore** value to 0.

The following figure shows the results.

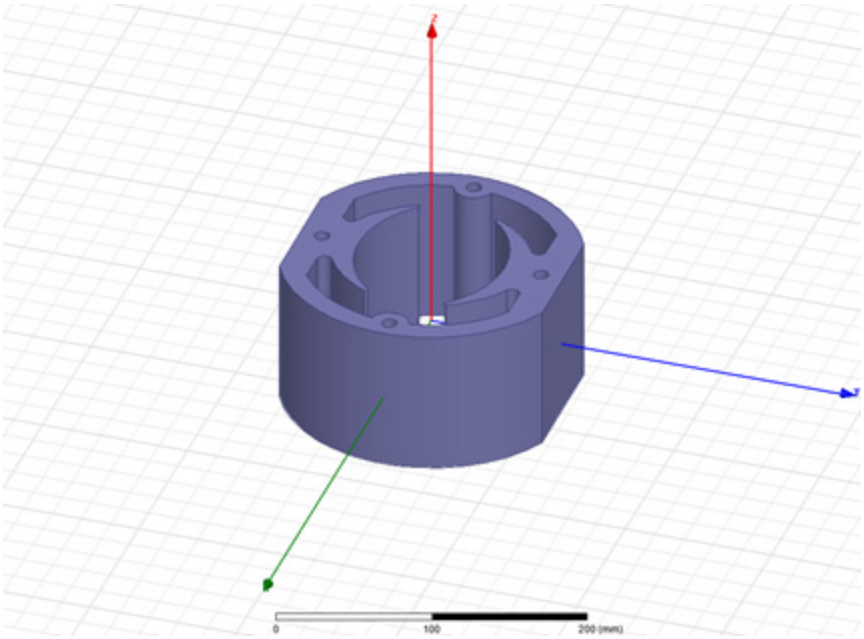


Figure 9-137 Outer universal-motor core

Creating All Coils (UnivMCore)

You can create all coils of a universal-motor core manually by setting the property of **InfoCoil** to 1.

You can assign the value of this property either when creating a universal-motor core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

The following figure gives an example of all coils for this UDP.

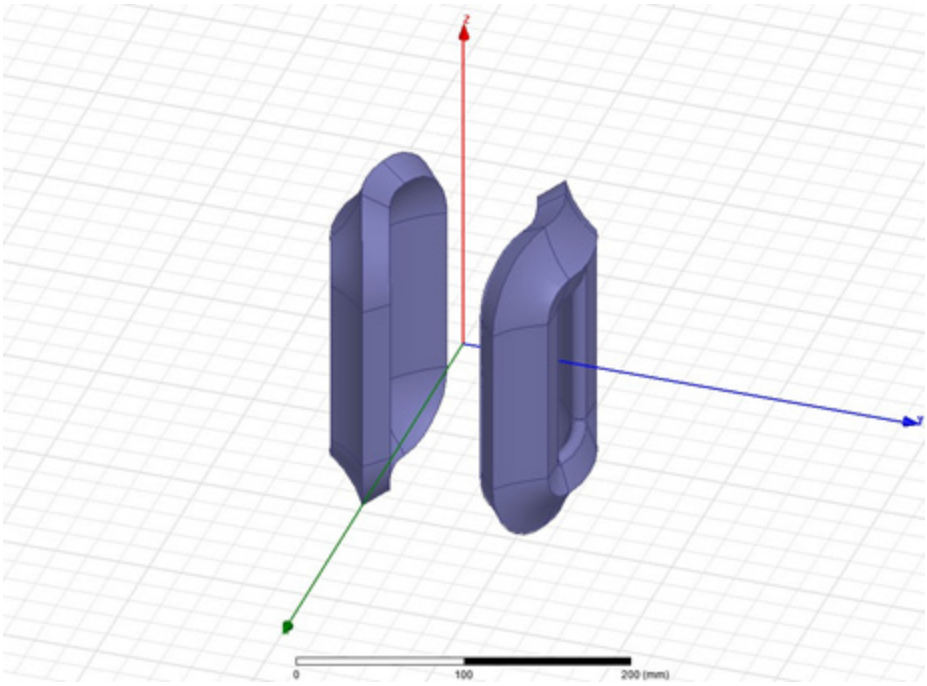


Figure 9-138 Coils of Universal-Motor Core

Creating a Coil and its Terminal (UnivMCore)

You can create a coil of a universal-motor core manually by setting the property of **InfoCoil** to 2, shown in Figure 9-136 as a red object.

You can create its terminals by setting the property of **InfoCoil** to 3 or 4 (for terminal1 and terminal2 respectively).Figure 9-139 shows what a terminal looks like.

You can assign the value of this property either when creating a universal-motor core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

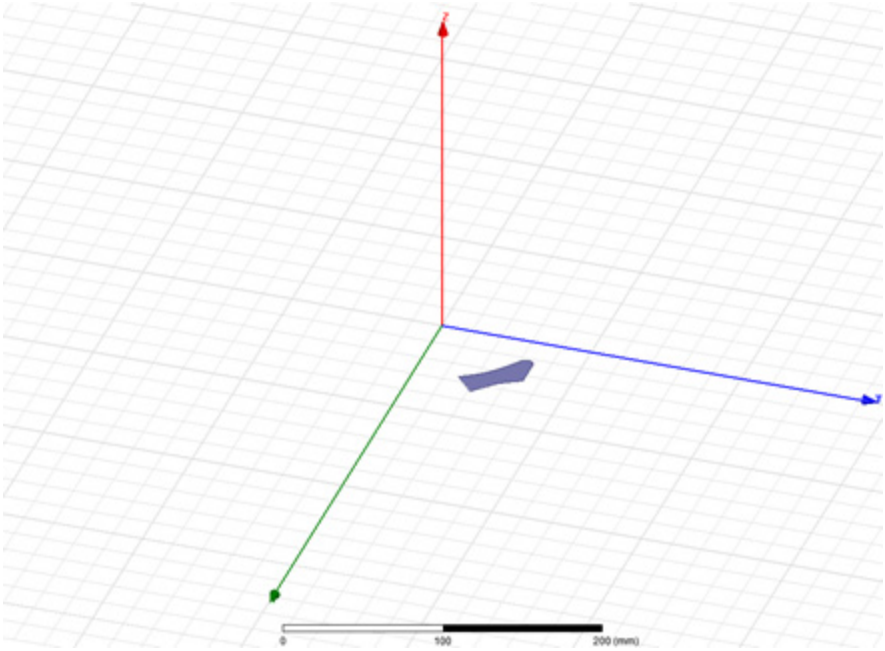


Figure 9-139 A terminal of a coil of Universal-Motor Core

SRMCore UDP

The SRMCore UDP is used to create the stator core, a stator winding, and the rotor core for switched reluctance motors. It can also create a coil terminal for current assignment.

Property	Description
DiaGap	Core diameter on gap side, DiaGap<DiaYoke for outer cores.
DiaYoke	Core diameter on yoke side, DiaYoke<DiaGap for inner cores.
Length	Core length.
Poles	Number of poles.
ThkYoke	Yoke thickness.
Embrace	Pole embrace (the ratio of pole arc to pole pitch).
Delta	One side pole-edge angle, 0 for parallel pole edges.
RFillet	RFillet > 0 for SRM cores with fillet corner; RFillet < 0, its absolute value defines the height of the V-shape slot bottom.
EndExt	Coil one-side end extended length.
LenRegion	Region length.
InfoCore	0: core; 1: core & coils; 2: coil; 3: terminal1; 4: terminal2; 100: region.

These parameters are used in the following figures.

Figure 9-142 shows the parameter **EndExt**, which is a parameter of a coil. The **InfoCore** value of **EndExt** is 1 (core & coils) to better describe the parameter.

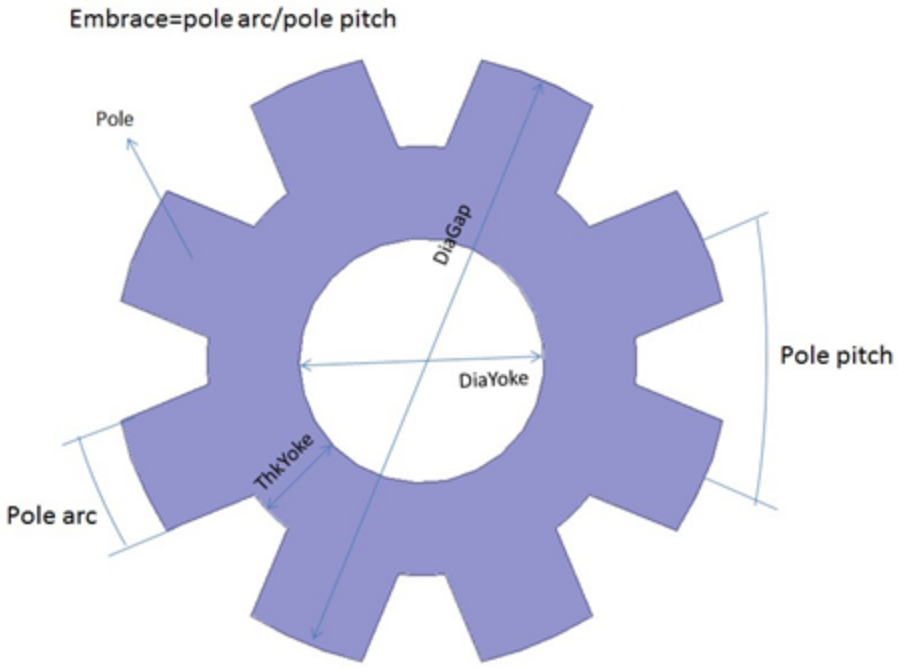


Figure 9-140 Inner core of SRM Core (InfoCore set to 0)

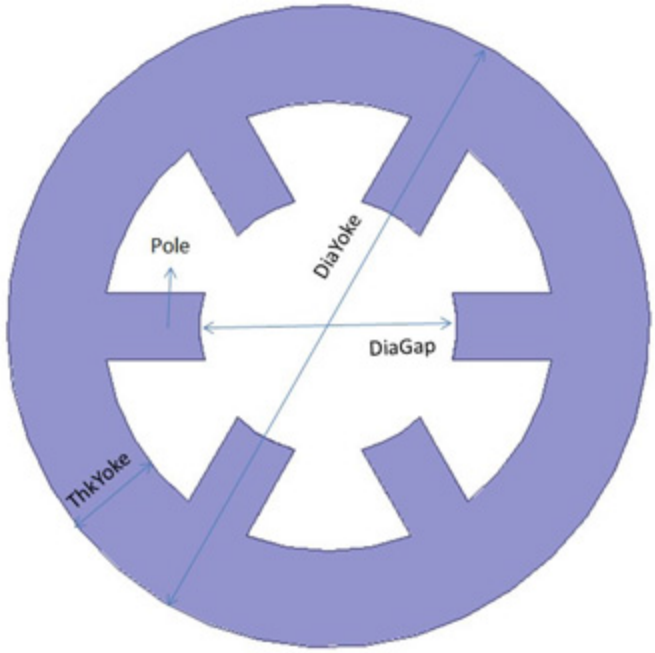


Figure 9-141 Outer core of SRM Core (InfoCore set to 0)

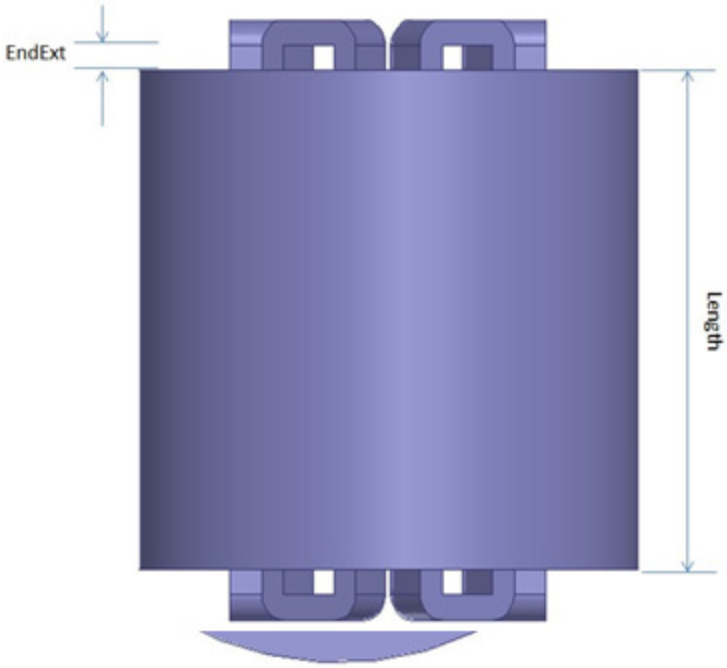


Figure 9-142 Core & coils of SRM Core (InfoCore set to 1)

Creating an Inner or Outer Core for Switched Reluctance Machines

You can set the value of **DiaYoke** and **DiaGap** as **DiaYoke**<**DiaGap** to create inner cores or **DiaGap**<**DiaYoke** for outer cores. You can set the values either when creating the core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Set the **InfoCore** value to 0.

Examples of inner SRM core are shown in Figure 9-140 and Figure 9-143 . For outer core, refer to Figure 9-141 and Figure 9-144 .

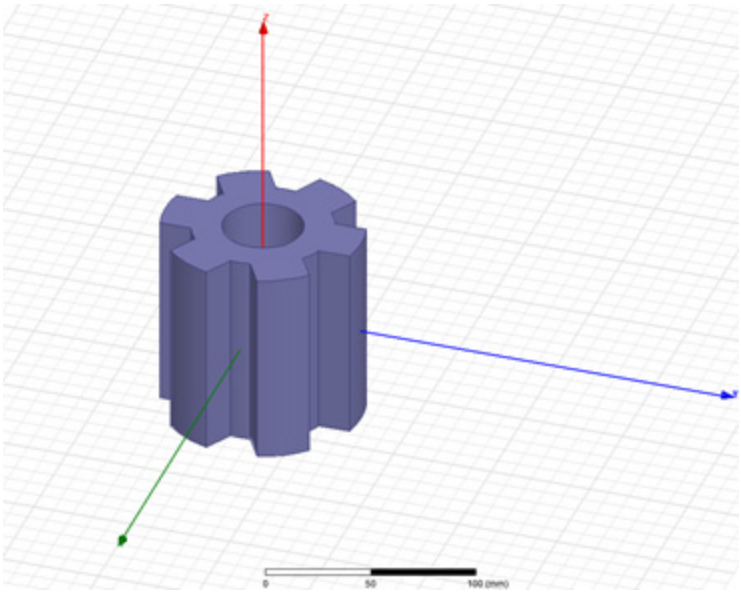


Figure 9-143 3D inner core of SRM Core (InfoCore set to 0)

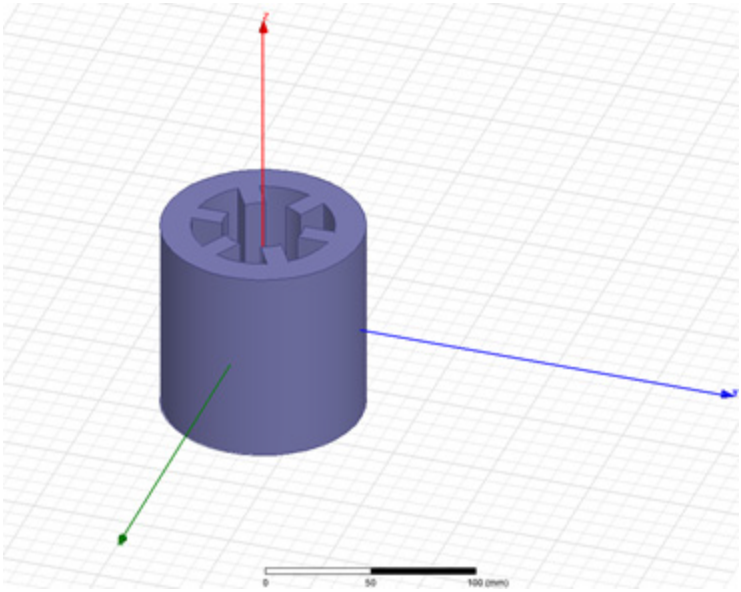


Figure 9-144 3D outer core of SRM Core (InfoCore set to 0)

Creating Core and All Coils (SRMCore)

You can create a core and all coils of an SRM Core manually by setting the value of **InfoCoil** to 1.

You can assign the value of this property either when creating an SRM core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Core and all coils can be created only when the core is an outer core (value of **DiaYoke** > **DiaGap**).

Examples of this core are shown in Figure 9-142 and Figure 9-145 .

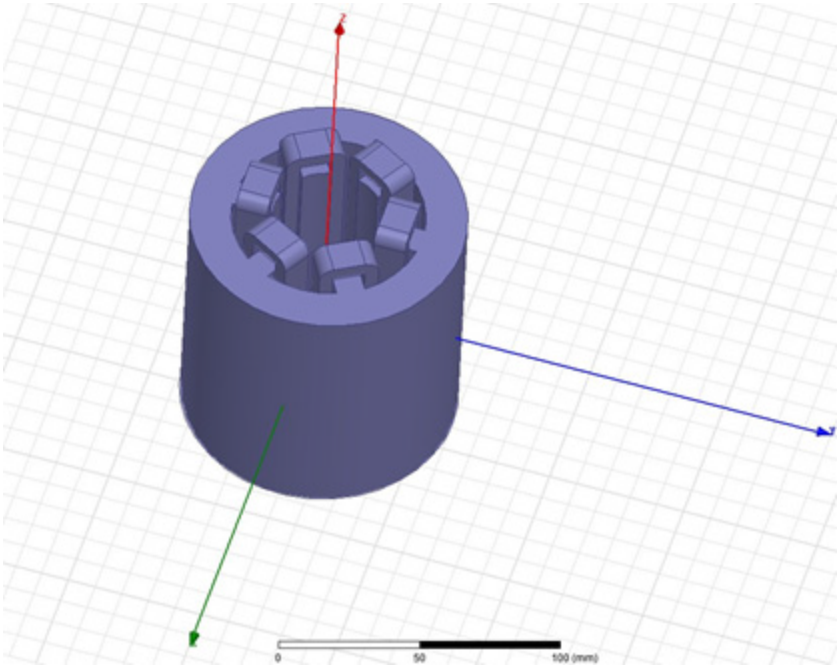


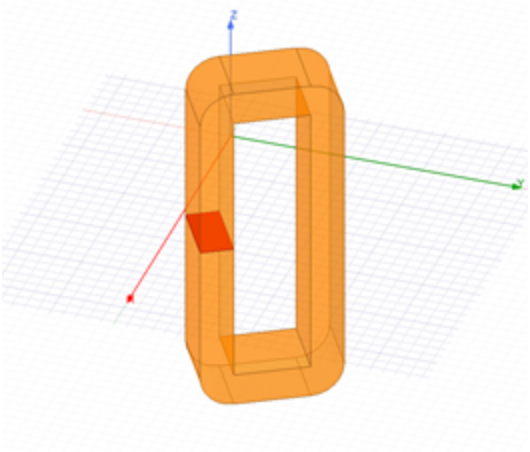
Figure 9-145 3D core and coils of SRM Core (InfoCore set to 1)

Creating a Coil and its Terminal (SRMCore)

You can create a coil of an SRM core manually by setting the value of **InfoCoil** to 2. An example is shown below.

You can create a terminal manually by setting the property of **InfoCoil** to 3 or 4, for terminal1 or terminal2 respectively. An example is shown by the red sheel in the figure.

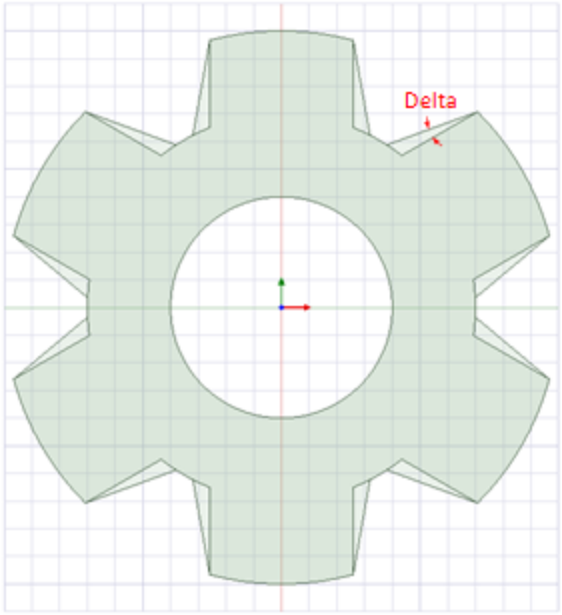
You can assign the value of this property either when creating an SRM core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.



Single coil and its terminal created by SRMCore

Creating Core with Unparallel Pole Edges (SRMCore)

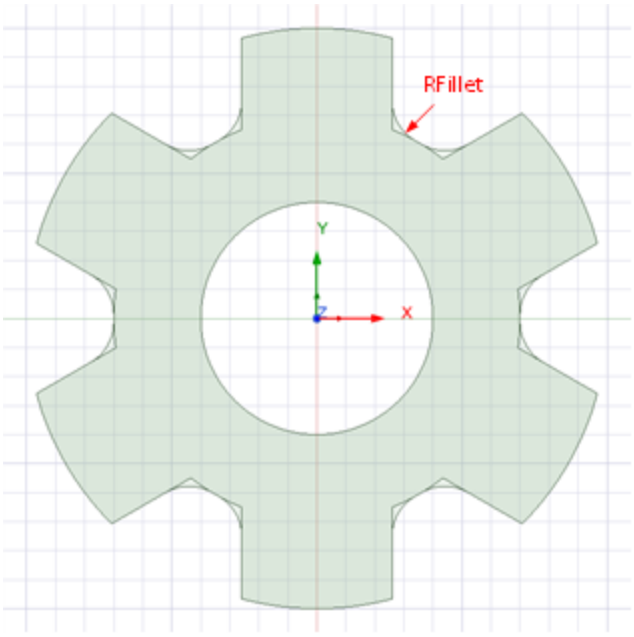
Parameter **Delta** is introduced to define one side pole-edge angle, as shown below. When **Delta** = 0, the two pole-edge sides are parallel.



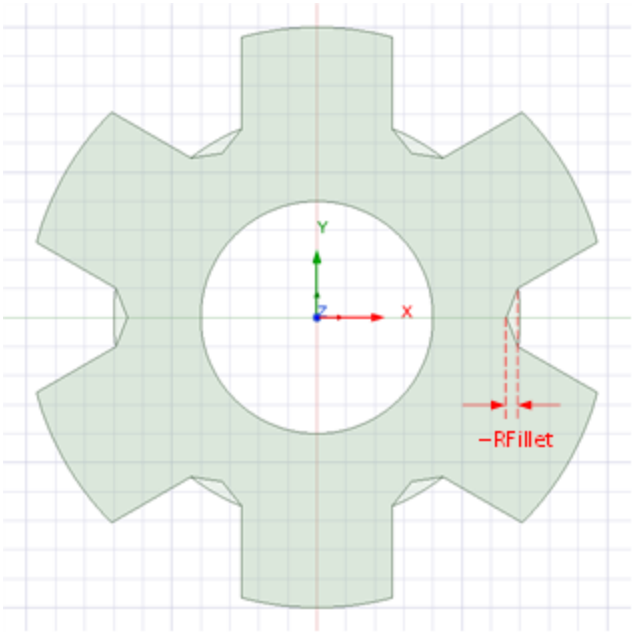
Parameter **Delta** for unparallel pole edges

SRM Core with Slot Corner Fillet and V-Shaft Bottom

Parameter **RFillet** is introduced to define corner fillet radius. **RFillet** > 0 for SRM cores with fillet corner as shown below.



When parameter **RFillet** < 0, its absolute value defines the height of the V-shape slot bottom as shown below.



SynRMCore UDP

The SynRMCore UDP is used to create the rotor core for synchronous reluctance machines.

Property	Description
DiaGap	Core diameter on gap side, or outer diameter.
DiaYoke	Core diameter on yoke side, or inner diameter.
Length	Core length.
Poles	Number of poles
PoleType	1: ALA; 2: arc; 3: hyperbolic; 4: hyperbolic line.
Barriers	Barriers per Pole, for PoleType 2 & 3 only.
H	Bridge thickness, for PoleType 2 & 3 only.
W	Rib width, for PoleType 2 & 3 only.
R	Barrier fillet radius, for PoleType 2 & 3 only.
R0	Radius of the bottom barrier arch, for PoleType 2 only.
Rb	Barrier bottom minimum radius.
Y0	Yoke bottom thickness.
B0	Barrier bottom thickness, for PoleType of 2 & 3 only.
LenRegion	Region length.
InfoCore	0: core; 1: one barrier; 100: region.

The SynRM UDP supports four pole types:

- **Type 1** – axially laminated core
- **Type 2** – barriers with concentric arcs and lines
- **Type 3** – barriers with hyperbolic curves
- **Type 4** – barriers with hyperbolic polylines

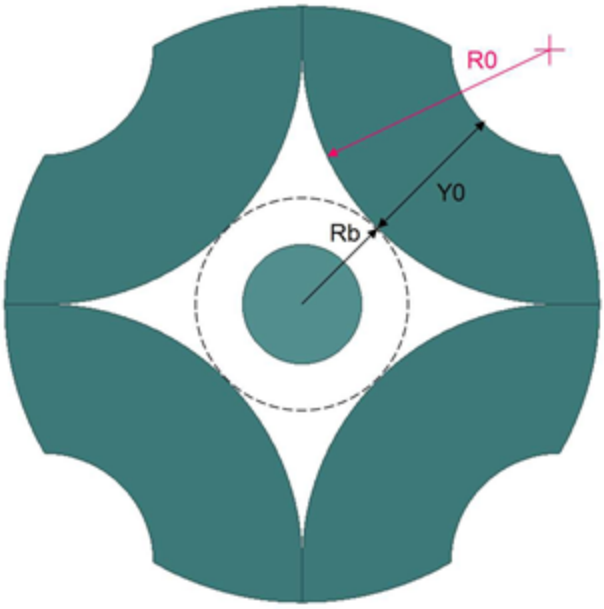
Type 1

Type 2

Type 3

Type 4

Parameters for SynRM core with PoleType = 1



In the figure, *R0* is a derived parameter, as given below:

$$R0 = Rb\sin\theta/(1 - \sin\theta)$$

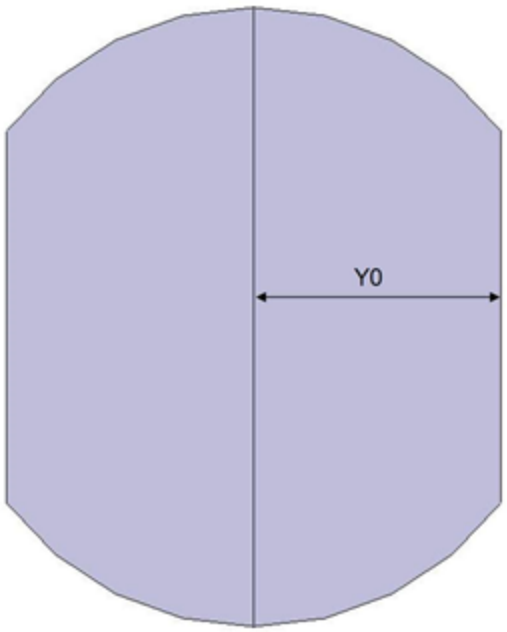
where $\theta = \pi/p$, and p, the number of poles, should be ≥ 4.

The center point (x, y) of the *R0* circle is:

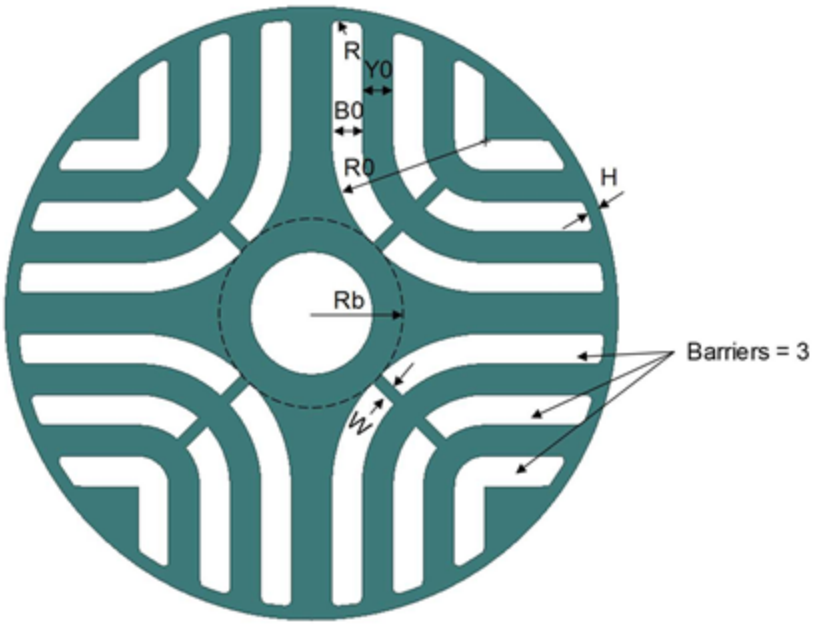
$$x = (Rb + R0)\cos\theta$$

$$y = (Rb + R0)\sin\theta$$

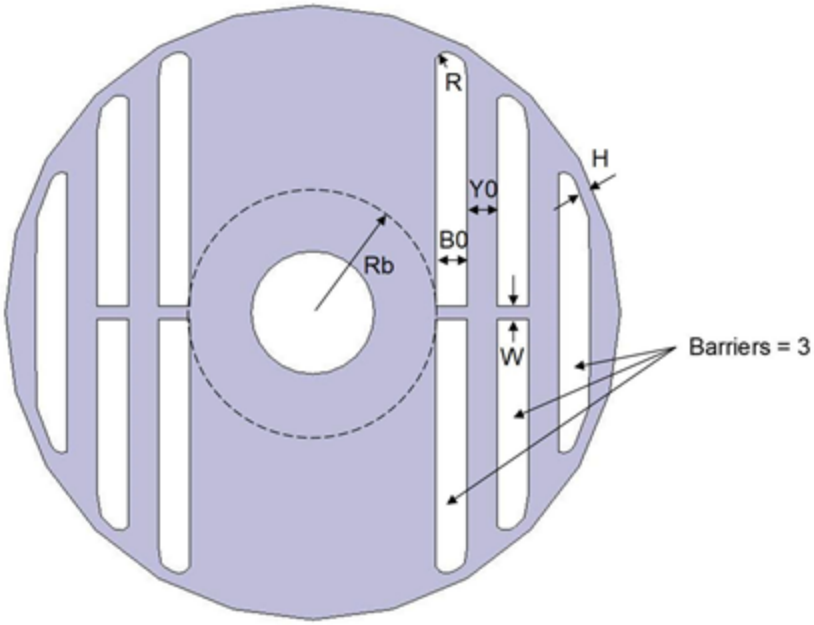
Below is a two-pole SynRM core with PoleType = 1.



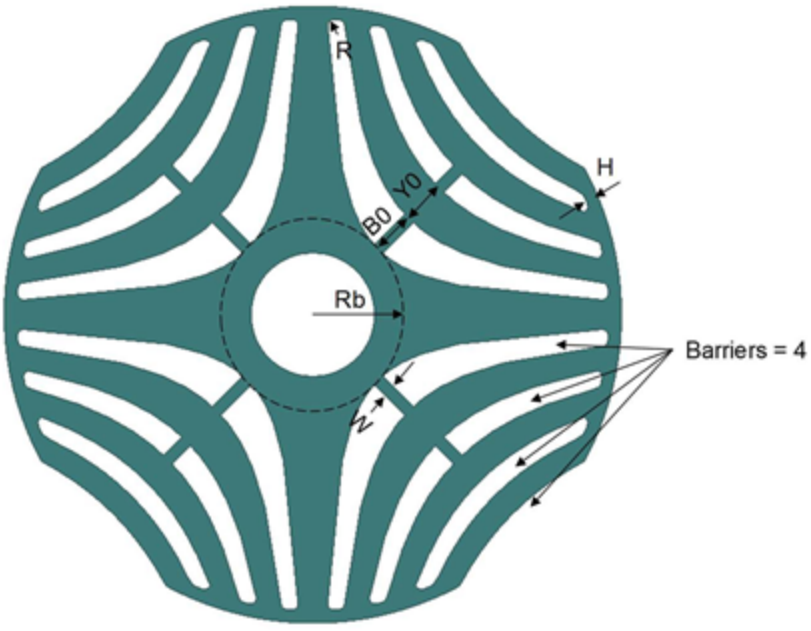
Parameters for SynRM core with PoleType = 2



Below is a two-pole SynRM core with PoleType = 2



Parameters for SynRM core with PoleType = 3



The main edges of barriers for Type 3 are hyperbolic curves defined by

$$r^2 \sin(pp \bullet \theta) = R_m^2$$

$$x = r \bullet \cos\theta$$

$$y = r \bullet \sin \theta$$

where *pp* is the number of pole pairs, $\pi/pp > \theta > 0$, and R_m is the minimum radius of a hyperbolic line. For different R_m values, you will get different hyperbolic curves.

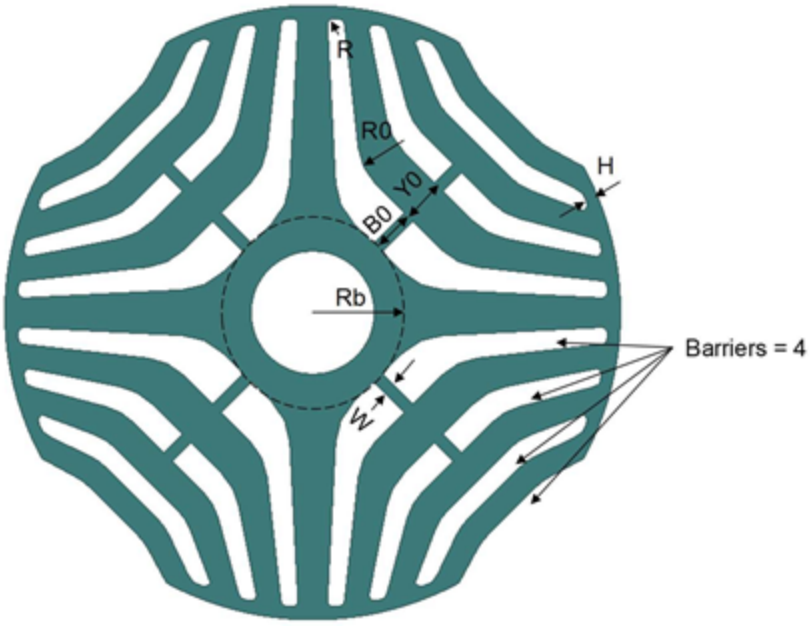
The first barrier is created based on the following two hyperbolic curves:

- for the bottom hyperbolic curve, *Rm2* = *Rb*
- for the top hyperbolic curve, *Rm1* = *Rb* +*B0*

The second and all other barriers are created based on a constant minimum barrier width and minimum tooth width at maximum limit radius which is derived from outer radius minus the value of bridge thickness parameter H.

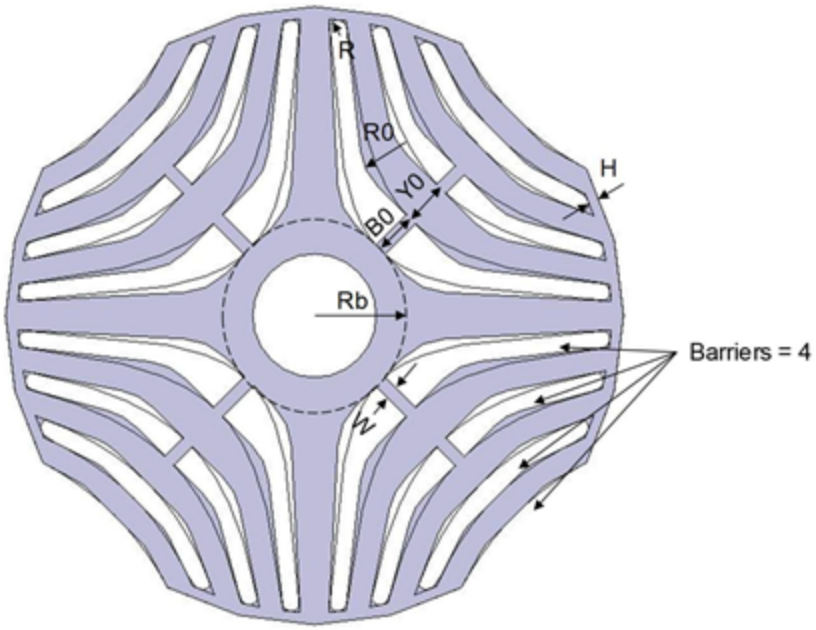
The minimum barrier width and minimum tooth width can be derived from *B0* and *Y0*, respectively. If the top and bottom hyperbolic curves are defined by *Rm1* and *Rm2*, respectively, and let *r* to be the maximum limit radius, you can obtain two points (*x1*, *y1*) and (*x2*, *y2*) at the maximum limit circle intersected with the top and bottom hyperbolic curves, respectively, based on the above hyperbolic curve equation. Please note that the distance between these two intersection points is not the minimum width because the connection line is not perpendicular to the center hyperbolic curve. The angle between the connection line and normal line is the angle between the *r* line and the tangent line which can be computed from *dy*/*dx* and *θ* of the center hyperbolic curve at the maximum limit radius.

Parameters for SynRM core with PoleType = 4



The main edges of barriers for Type 4 are hyperbolic polylines. A hyperbolic polyline is derived from a hyperbolic curve. It consists of one bottom line (tangent to the bottom points of the hyperbolic curve), two side lines (tangent to the point on the hyperbolic curve at the maximum limit radius), and fillet arcs defined *R0*. The value of fillet radius *R0* is the same for all hyperbolic

polylines. All parameters for hyperbolic curves to derive hyperbolic polylines are defined in the same way as those for pole Type 3. The following figure shows how barriers of Type 4 are derived from those of Type 3.



Note: Two-pole SynRM cores for Types 3 and 4 will be the same as those of Type 2.

Creating an Inner Core for a Synchronous Reluctance Machine

You can set the value of **DiaYoke** and **DiaGap** as **DiaYoke<DiaGap** to create inner cores. You can set the value either when creating the core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Set the **InfoCore** value to 0.

Figure 9-146 gives an example of an inner core for SynRM.

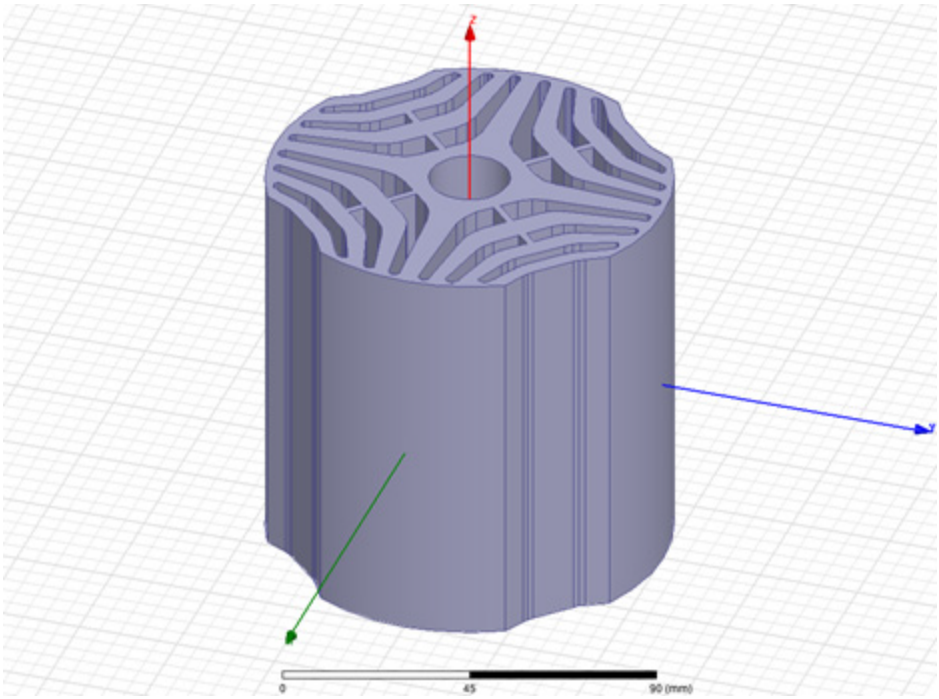


Figure 9-146 3D SynRM Core

Creating a SynRMCore with Different Barriers

You can create a SynRMCore with different barriers through several steps, as follows.

1. Create a SynRMCore as shown in Creating an Inner Core for a Synchronous Reluctance Machine.

2. Change the value of Barriers to 1. The result is shown in the following figure.

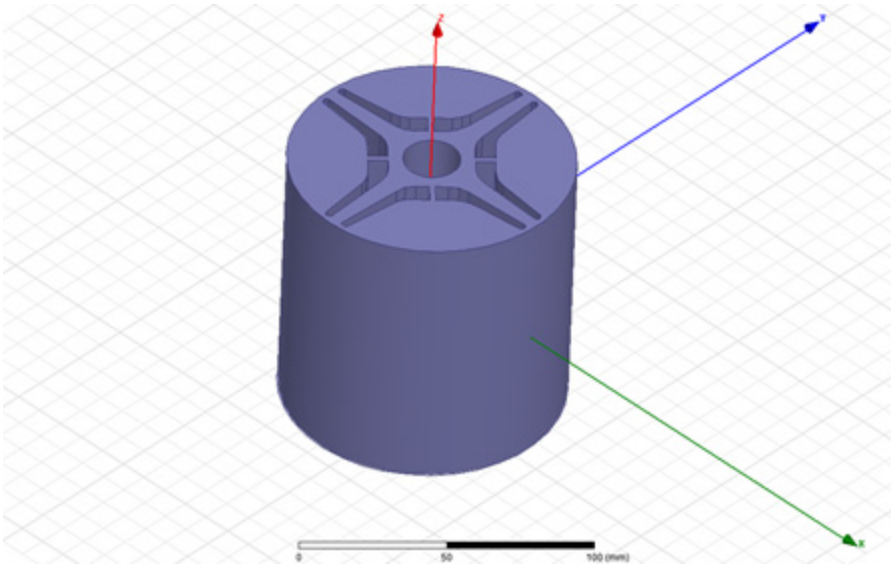


Figure 9-147 A SynRMCore with one barrier

3. Change the value of **DiaGap** to be larger and make the value of **Y0** smaller to make enough space for other barriers. You can modify these values by clicking on **CreateUserDefinedPart** in the history tree under the SynRMCore, as shown in the following figure.

   As an example, the values of **DiaGap** and **Y0** are changed to 130 and 5, respectively.

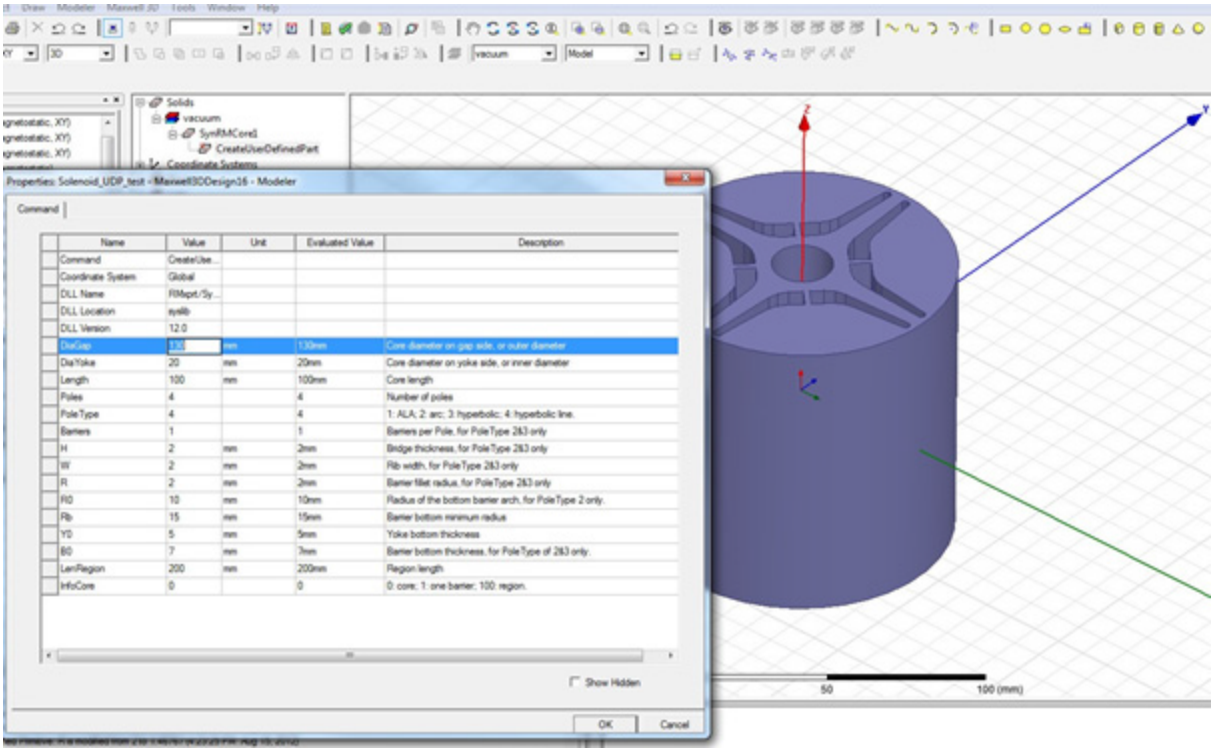


Figure 9-148 Changing the values of DiaGap and Y0

4. Make a copy (or multiple copies if more than 1 barrier is needed) of the current SynRMCore object.

   In the history tree, click **SynRMCore1**, press **Ctrl**+**C**, and then press **Ctrl**+**V** to duplicate the SynRMCore. After these processes, the history tree looks like the following figure (only for 1 copy situation):

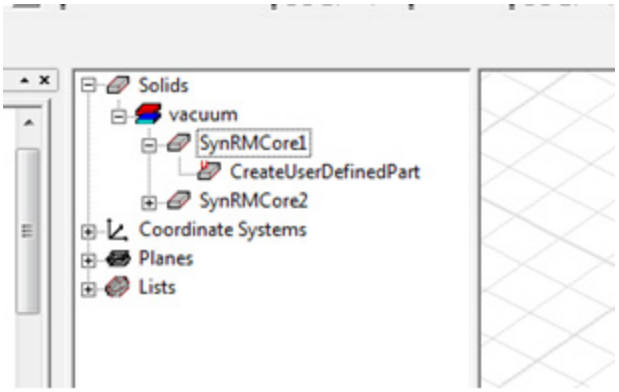


Figure 9-149 Make a copy of SynRMCore

5. Change the value of **InfoCore** of SynRMCore2 to 1 to create a barrier.

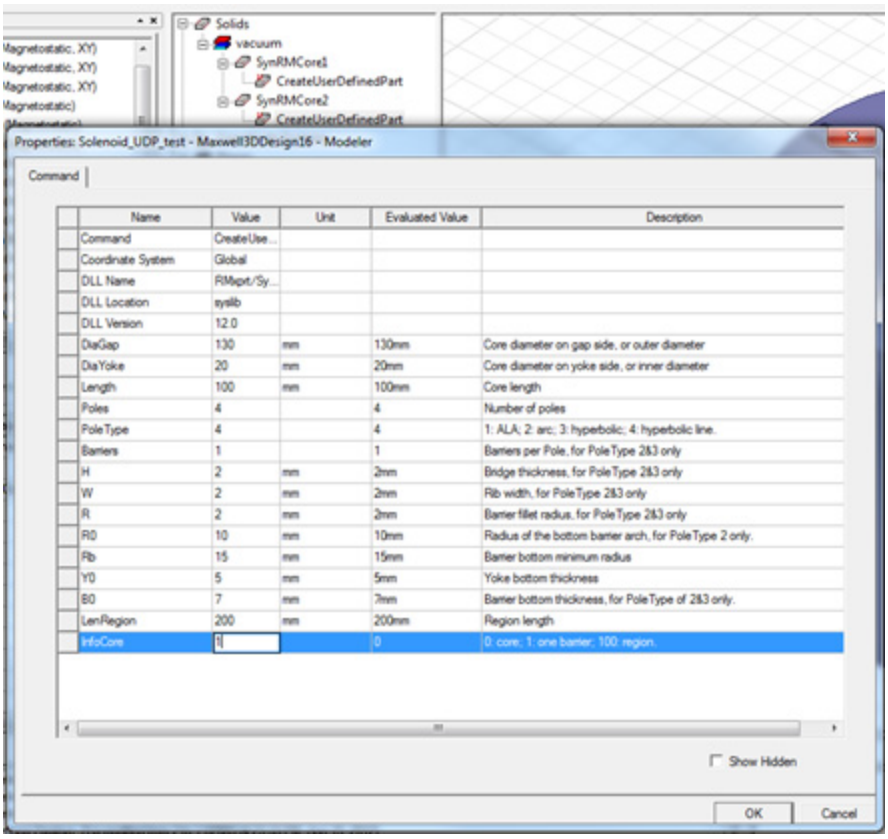


Figure 9-150 Changing the value of InfoCore of SynRMCore2

The result of this change is shown in the following figure.

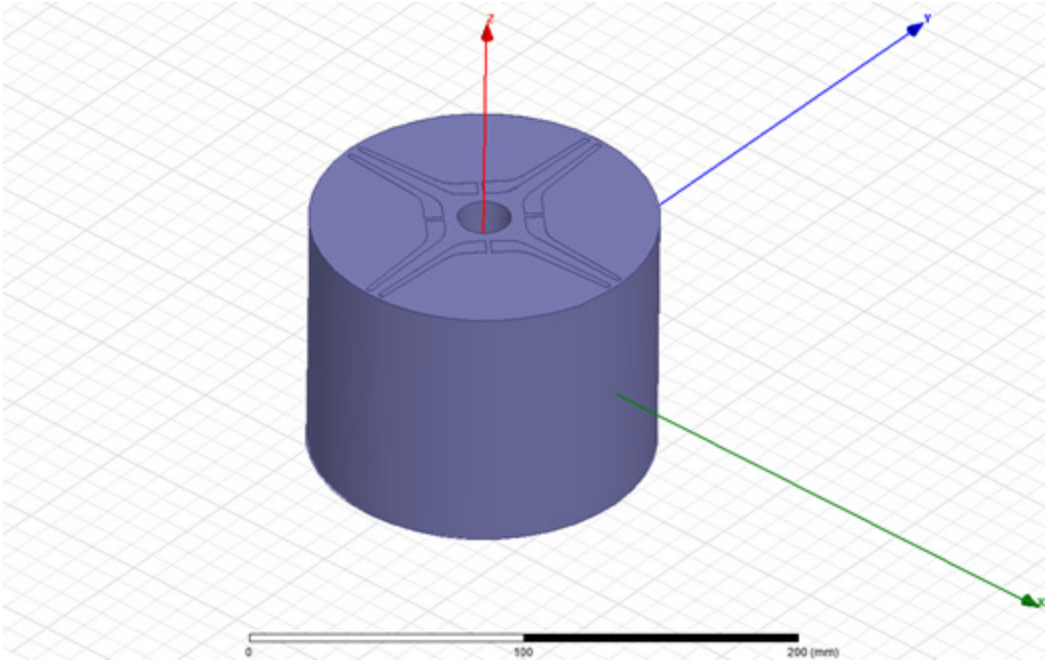


Figure 9-151 Result of InfoCore of SynRMCore2 being changed

6. Change the value of **Rb** of SynRMCore2 (barrier) to make it larger, and thus make the barrier away from the center of the core.

   As an example, the value of **Rb** of SynRMCore2 is changed from 15 to 25 by clicking on **CreateUserDefinedPart** in the history tree under the SynRMCore2, as shown in the following figure.

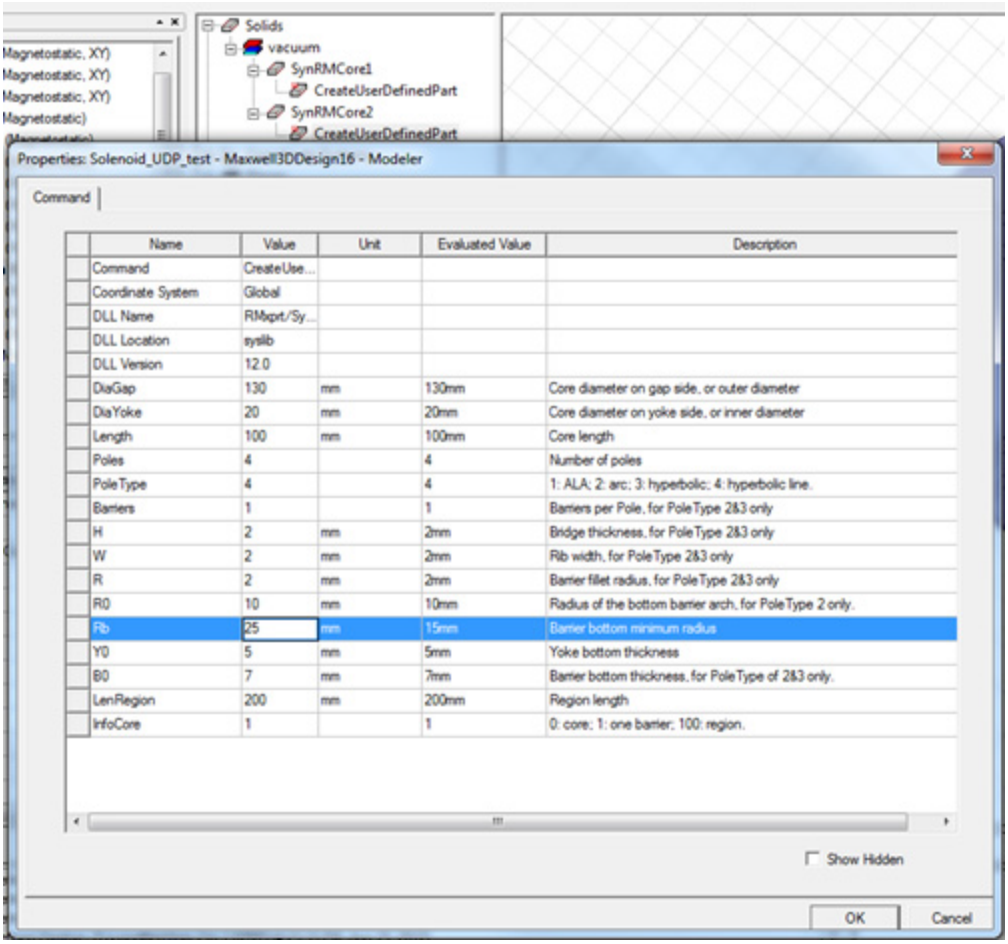


Figure 9-152 Changing the value of Rb of SynRMCore2

Another option is to change other values of SynRMCore2 (barrier) according to different needs, such as set the value of **W** to 0.

7. Then rotate the coordinate system. Objects in it should look like those in the following figure:

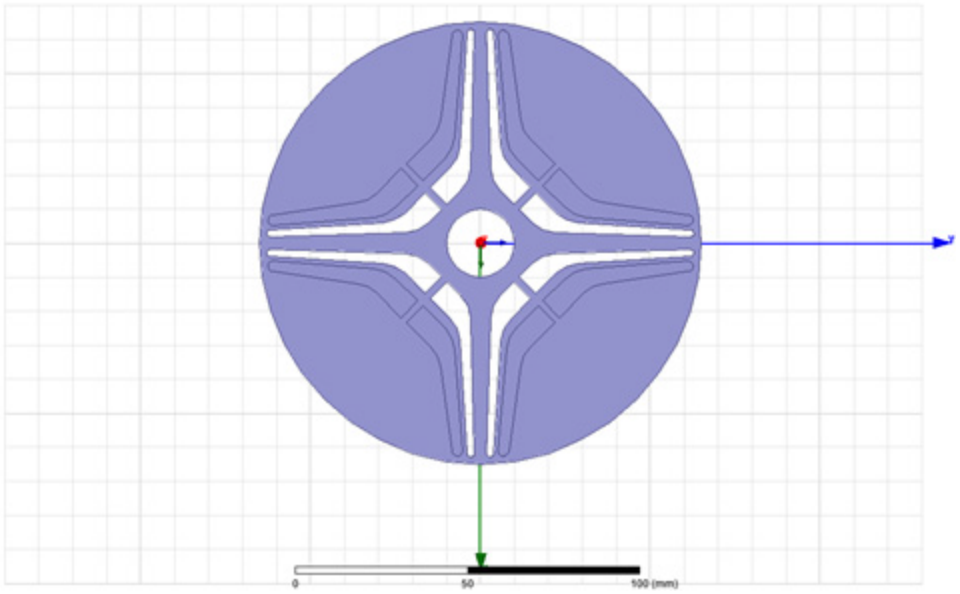


Figure 9-153 Before subtraction of two objects

8. Make a subtraction for two objects

   Choose both objects simultaneously by pressing **Ctrl** and clicking on two objects in the history tree on the screen.

   Then click **Modeler>Boolean>Subtract** to do the subtraction.

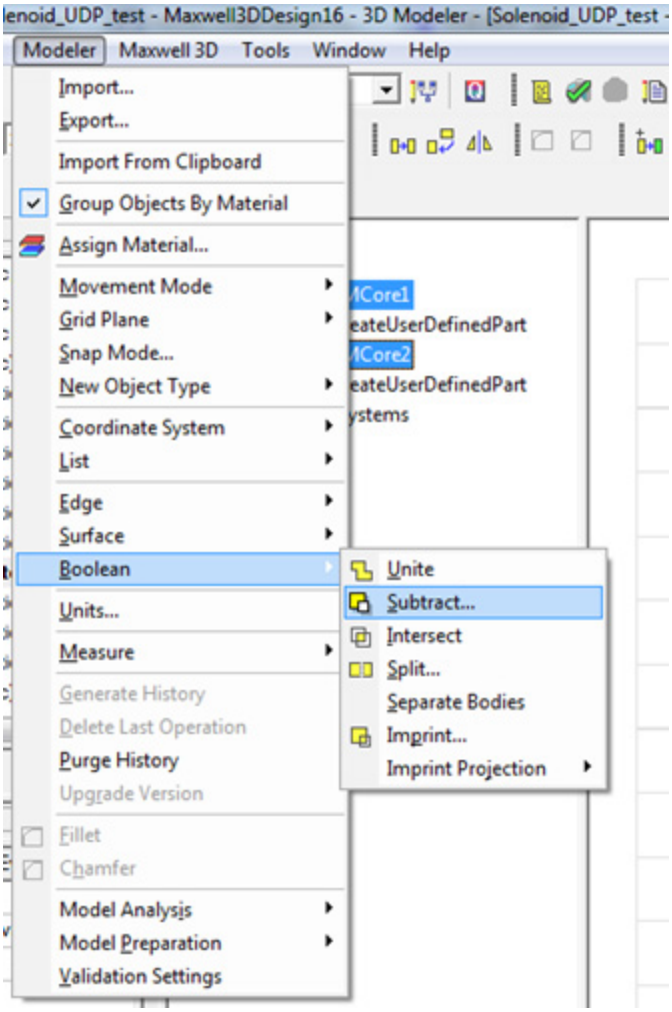


Figure 9-154 Make a subtraction

The SynRMCore1 should be in **Blank Parts** whereas SynRMCore2 is in **Tool Parts**, as shown in the follow figure.

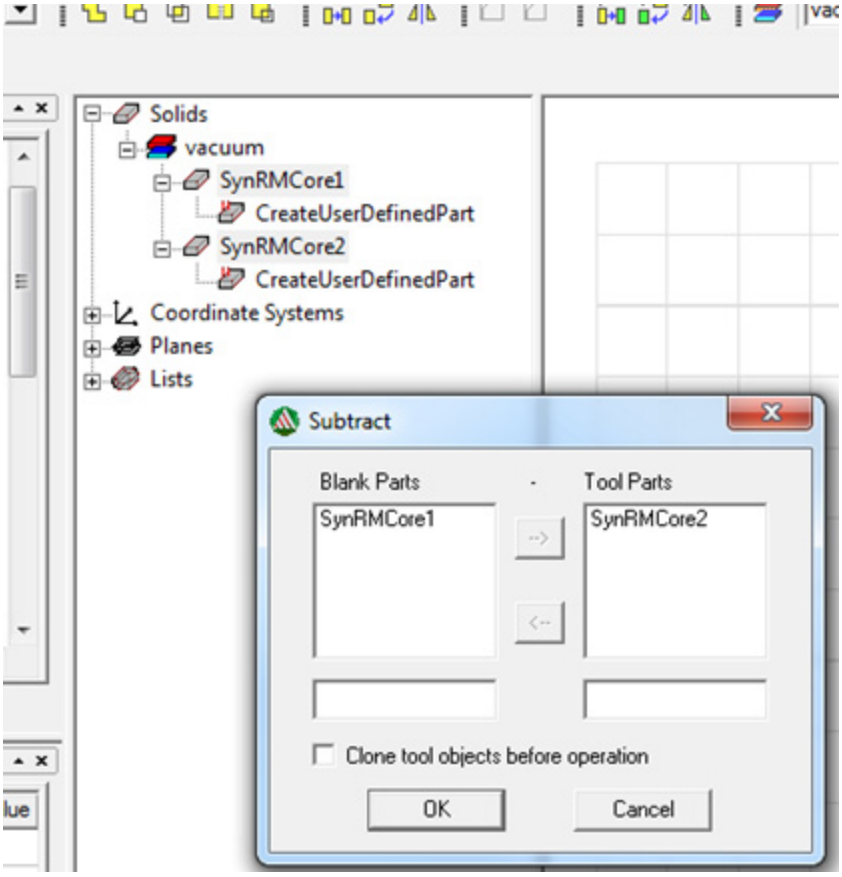


Figure 9-155 Choosing Blank Parts and Tool Parts

9. Click **OK** to finish subtraction. A SynRMCore with different barriers is shown in the following figure.

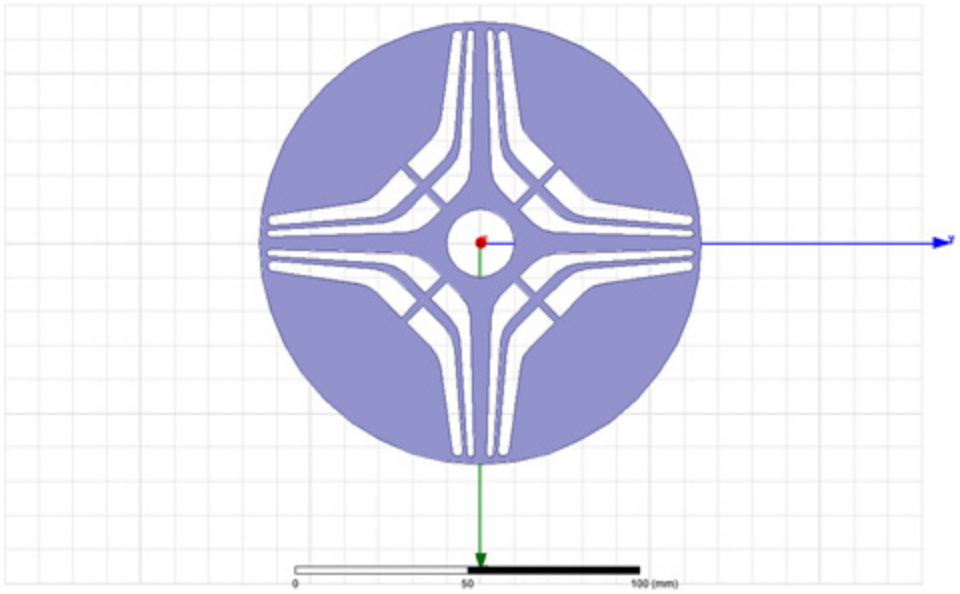


Figure 9-156 After subtraction

To make the barriers look different, you can change the parameters of barriers like **W** and **B0**. Also, repeating the processes can make multiple barriers.

ClawPoleCore UDP

The ClawPoleCore UDP is used to create the claw-pole rotor core, a field winding and/or a PM for claw-pole alternators. It can also create a coil terminal for current assignment.

Property	Description
DiaGap	Core diameter on gap side, DiaGap<DiaShaft for outer cores.
DiaShaft	Core diameter on shaft side, DiaShaft<DiaGap for inner cores.
Length	Core length.
Poles	Number of poles.
EmbraceTip	Embrace of pole tip.
EmbraceRoot	Embrace of pole root.
ThickTip	Thickness of pole tip.
ThickRoot	Thickness of pole root.
ThickShoe	Thickness of side shoes.
DepthSlot	Depth of slot between two poles.
ThickYoke	Thickness of yoke.

Property	Description
LengthPole	Length of pole from tip to tip.
LengthMag	Length of magnet or the second air-gap.
SegAngle	Deviation angle for skewed pole sides (1~10, <1 for true surface).
LenRegion	Region length.
InfoCore	0: core; 1: core & coil; 2: coil; 3: magnet; 4: terminal; 100: region.

These parameters are used in the following figures. Figure 9-158 is in transparent to show parameters **EmbraceTip** and **EmbraceRoot** which otherwise are hard to show. For **LengthMag** parameter, refer to Figure 9-163 .

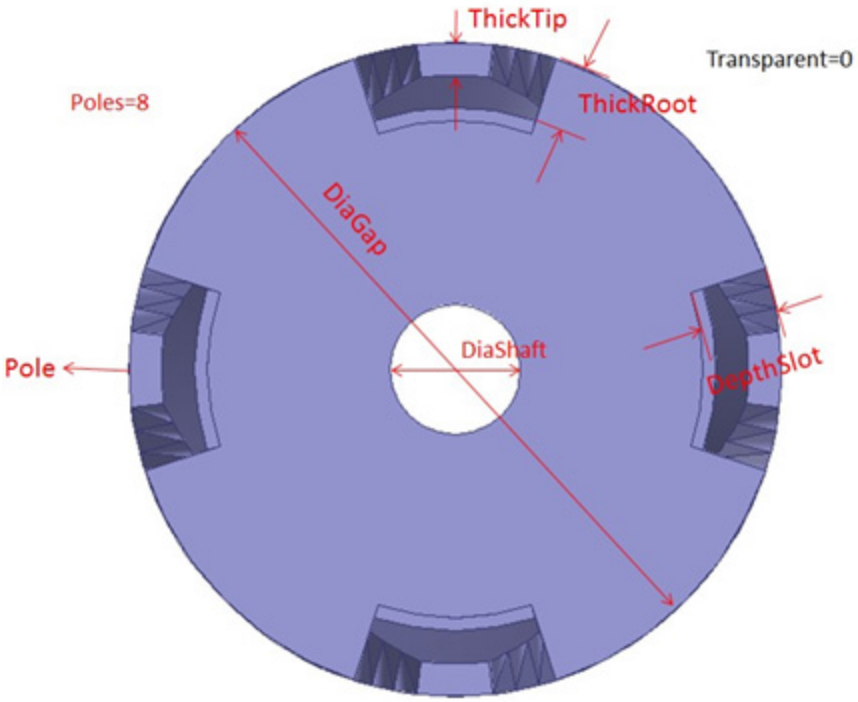


Figure 9-157 Claw Pole Core (InfoCore set to 0)

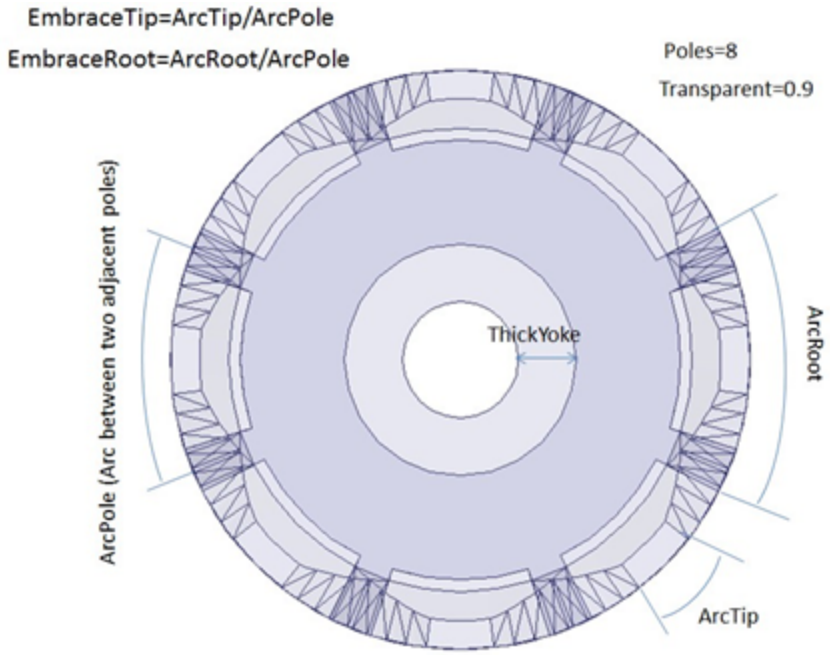


Figure 9-158 Transparent Claw Pole Core (InfoCore set to 0)

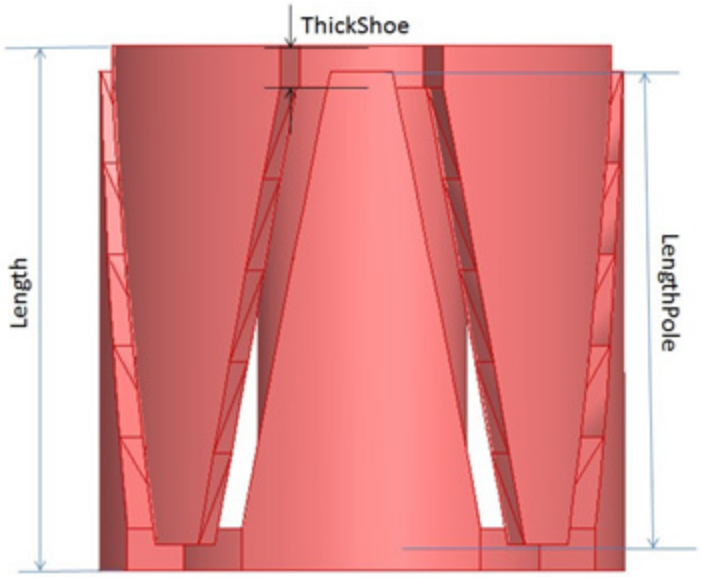


Figure 9-159 3D Claw Pole Core (InfoCore set to 0)

Creating an Inner Claw Pole Core

Set the value of **DiaShaft** and **DiaGap** as **DiaShaft<DiaGap** to create an inner claw pole core. You can set the value either when creating the core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Set the **InfoCore** value 0.

Examples of inner claw-pole core are shown in Figure 9-157 , Figure 9-158 , and Figure 9-159 .

Creating a Coil and its Terminal (Claw Pole)

You can create a coil of a claw pole core manually by setting the value of **InfoCoil** to 2. An example is shown inFigure 9-160 .

You can create a terminal of a claw pole core manually by setting the value of **InfoCoil** to 4. Figure 9-161 gives an example of this terminal.

You can assign the value of this property either when creating a claw-pole core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

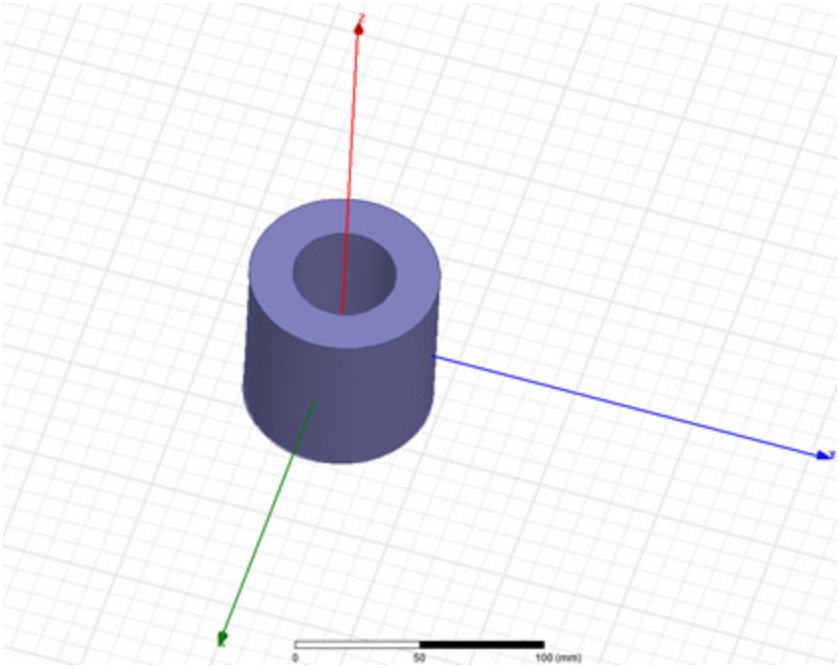


Figure 9-160 Coil of a claw-pole core

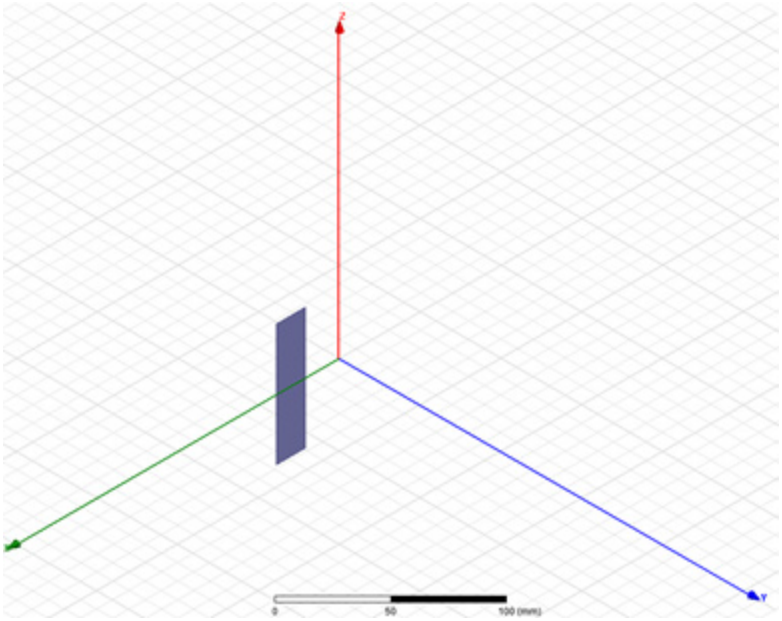


Figure 9-161 A terminal of coil in this claw-pole core

Creating a Magnet (Claw Pole)

To create a magnet, you need to set the value of **LengthMag** larger than 0 (by default, it is 0) and value of **InfoCore** to be 3. An example is shown in the following figure.

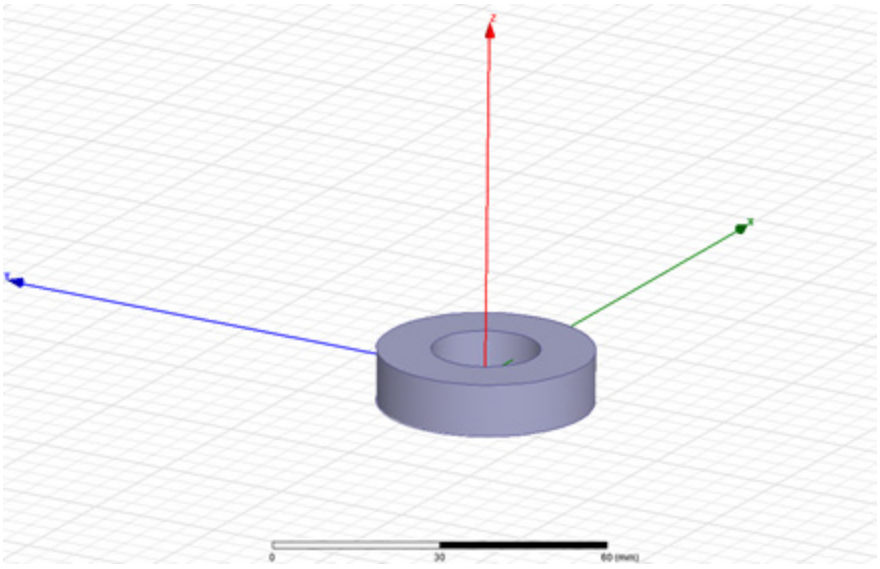


Figure 9-162 A magnet of ClawPoleCore

The following figure visualizes the **LengthMag** parameter.

Figure 9-163 LengthMag of a magnet of ClawPoleCore

You can assign the value of these properties either when creating a claw-pole core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

PMCore UDP

The PMCore UDP is used to create a PM stator/rotor core and a PM for PM commutating DC machines, PM adjustable-speed synchronous machines, or PM brushless DC motors.

Property	Description
DiaGap	Core diameter on gap side, DiaGap<DiaYoke for outer cores.
DiaYoke	Core diameter on yoke side, DiaYoke<DiaGap for inner cores.
Length	Core length.
Poles	Number of poles.
PoleType	Pole type: 1 to 5.
Embrace	Pole embrace (not for type 4).
ThickMag	Max thickness of magnets.
WidthMag	Magnet width (for types 4 & 5).
Offset	Pole arc offset (for types 1, 2 & 3).
Bridge	Bridge thickness (for type 5 only).
Rib	Rib width (for type 5 only), Rib=0 for rectangle ducts.
LenRegion	Region length.
InfoCore	0: core; 1: magnets; 2: magnet; 100: region.

These parameters are used in the following figures:

Figure 9-165 shows the **Offset** parameter when value of **PoleType** is 3.

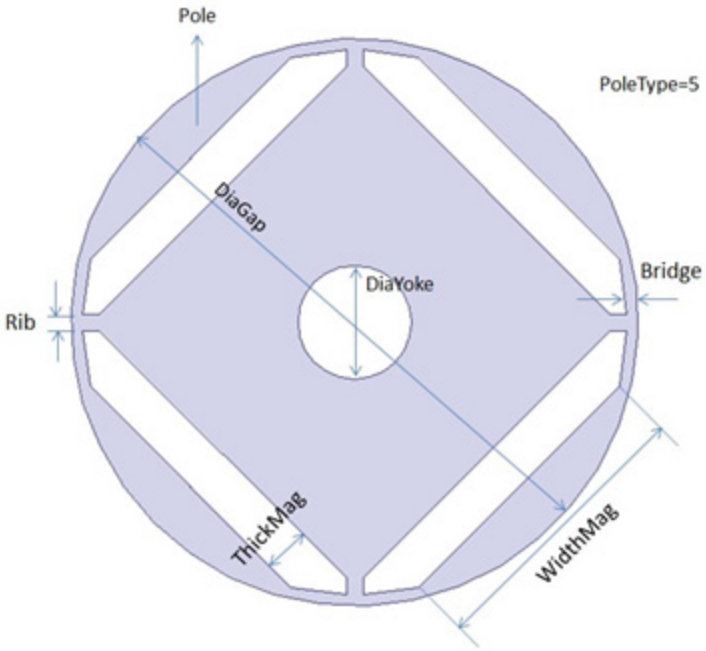


Figure 9-164 PM Core with PoleType set to 5 (InfoCore set to 0)

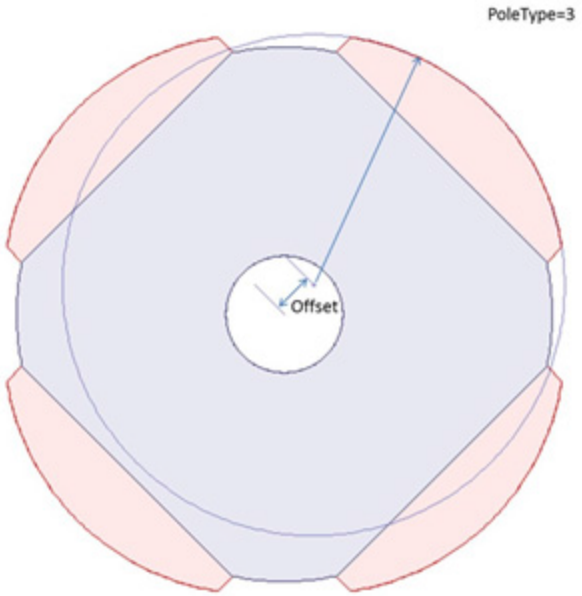


Figure 9-165 PM Core with PoleType set to 3 (InfoCore set to 0)

Creating an Inner or Outer PM Core

Set the value of **DiaYoke** and **DiaGap** as **DiaYoke**<**DiaGap** to create inner cores or **DiaGap**<**DiaYoke** for outer cores. In addition, r to create an outer core, the **Pole Type** must be 1 or 2. You can set the values either when creating the core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Set the **InfoCore** value to 0.

An inner PM core and an outer PM core are shown in Figure 9-166 and Figure 9-167 , respectively.

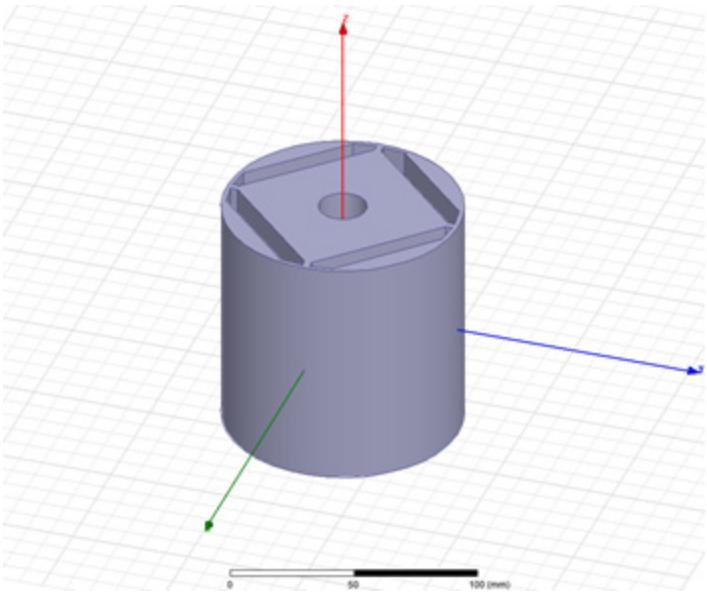


Figure 9-166 Inner PM core with PoleType set to 5 (InfoCore set to 0)

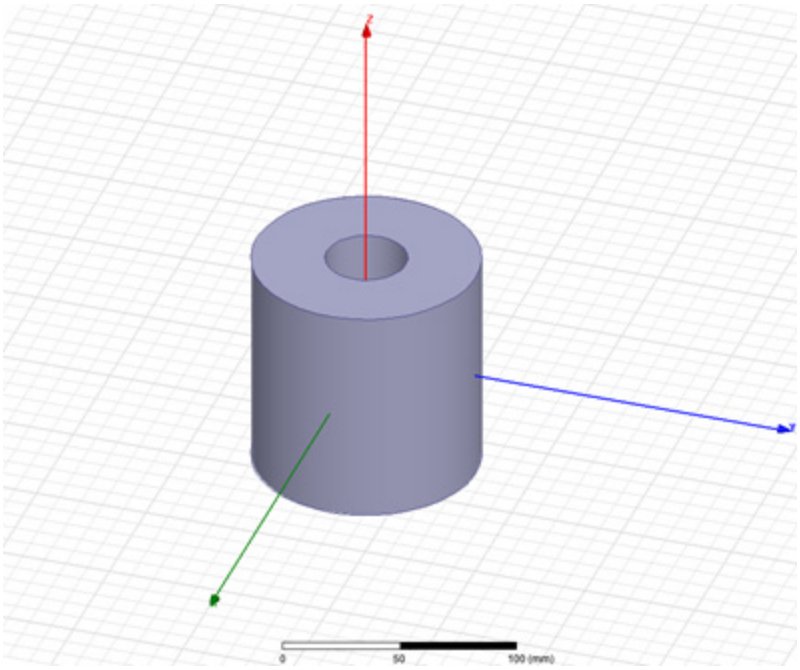


Figure 9-167 Outer PM core with PoleType set to 2 (InfoCore set to 0)

Creating All Magnets (PMCore)

You can create all magnets of a PM core manually by setting the value of **InfoCore** to 1. An example is shown by the four inner blue objects in the following figure, for which InfoCore has been set to 0 allowing the OuterPM core (gray object) and all magnets for PoleType=2 to be shown.

You can assign the value of this property either when creating a PM Core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

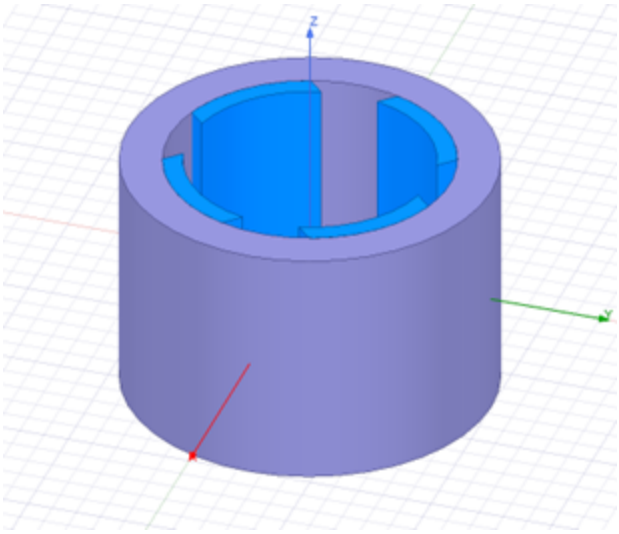


Figure 9-168 All magnets of a PM core

Creating a Magnet (PM Core)

You can create a magnet of a PM core manually by setting the value of **InfoCore** to 2. An example is shown in the following figure.

You can assign the value of this property either when creating a PM core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

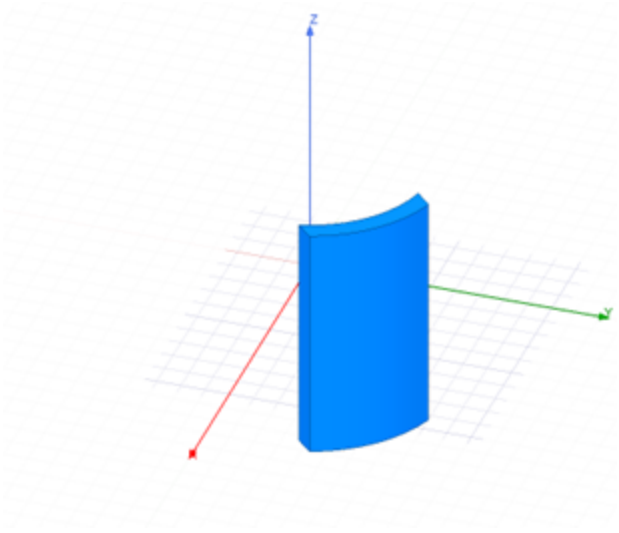
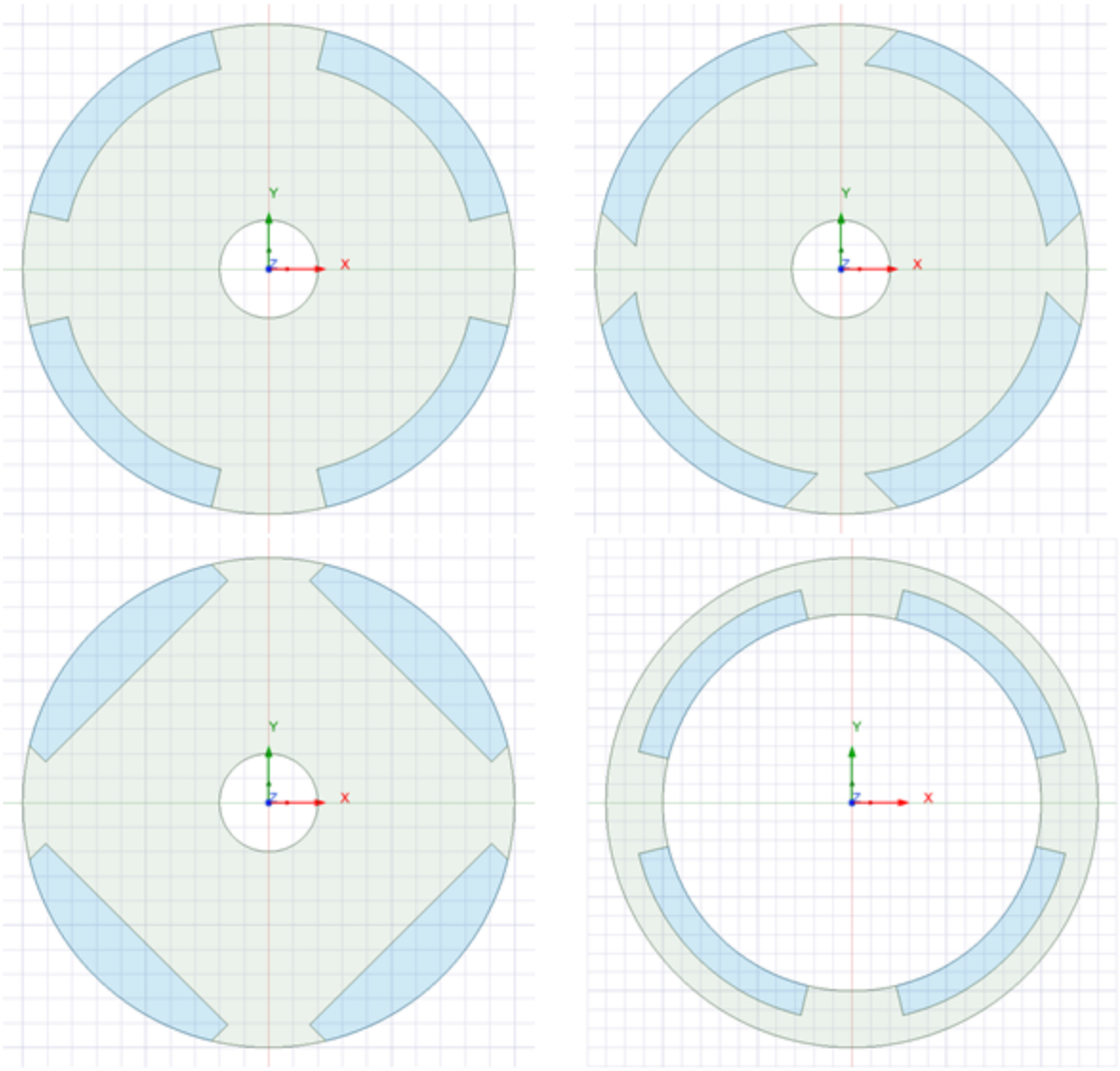


Figure 9-169 A magnet of PM Core

Creating an Insert Surface PM Core

When the value of **PoleType** is 1 to 3, you can create an insert surface PM core by setting **Embrace** to a negative value as shown in the examples below.



IPMCore UDP

The IPMCore UDP is used to create an IPM core and a PM for IPM synchronous machines or IPM brushless DC motors.

Property	Description
DiaGap	Core diameter on gap side, or outer diameter.
DiaYoke	Core diameter on yoke side, or inner diameter.
Length	Core length.
Poles	Number of poles.
PoleType	Pole type: 1 to 6.
D1	Limited diameter of PM ducts.
O1	Bottom width for separate or flat-bottom duct (for types 3~6).
O2	Distance from duct bottom to shaft surface (for types 2~6).
B1	Duct thickness.
Rib	Rib width.
HRib	Rib height (for types 3~5).
DminMag	Minimum distance between side magnets (for types 3~5).
ThickMag	Magnet thickness.
WidthMag	Total width of all magnet per pole.
LenRegion	Region length.
InfoCore	0: core; 1: magnets; 2: ducts; 3: one-pole magnet; 100: region.

Most parameters of IPMCore UDP are shown in the following figure except those common parameters like **DiaGap**, **DiaYoke** and **Length**.

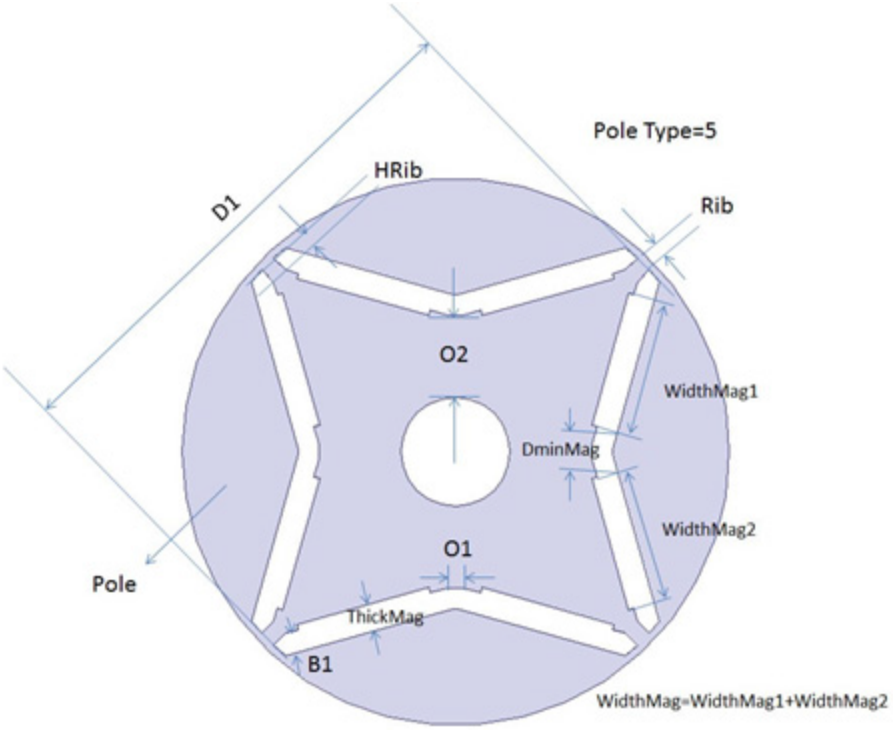


Figure 9-170 Parameters of IPM Core

Creating an Inner IPM Core

Set the value of **DiaYoke** and **DiaGap** as **DiaYoke<DiaGap** to create an inner IPM core. You can set the value either when creating the core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Set the **InfoCore** value to 0.

An example is shown in the following figure.

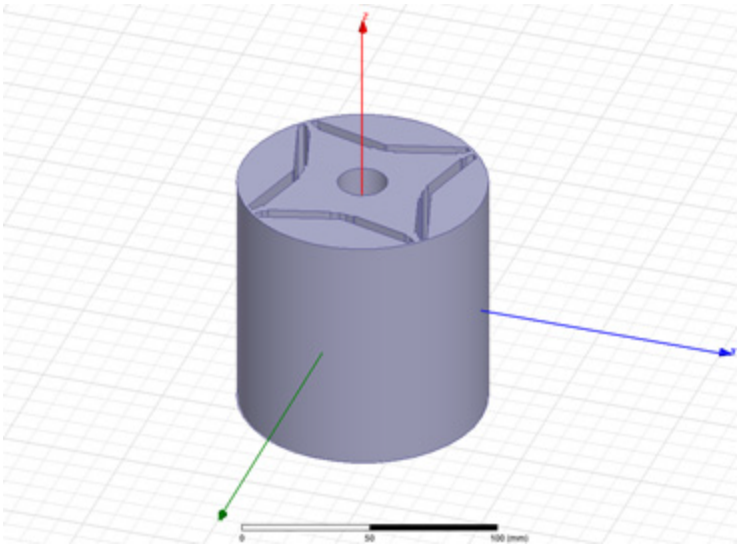


Figure 9-171 Inner IPM Core

Creating All Magnets (IPM Core)

You can create all magnets of an IPM core manually by setting the value of **InfoCoil** to 1.

You can assign the value of this property either when creating an IPM Core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

An example is shown in the following figure.

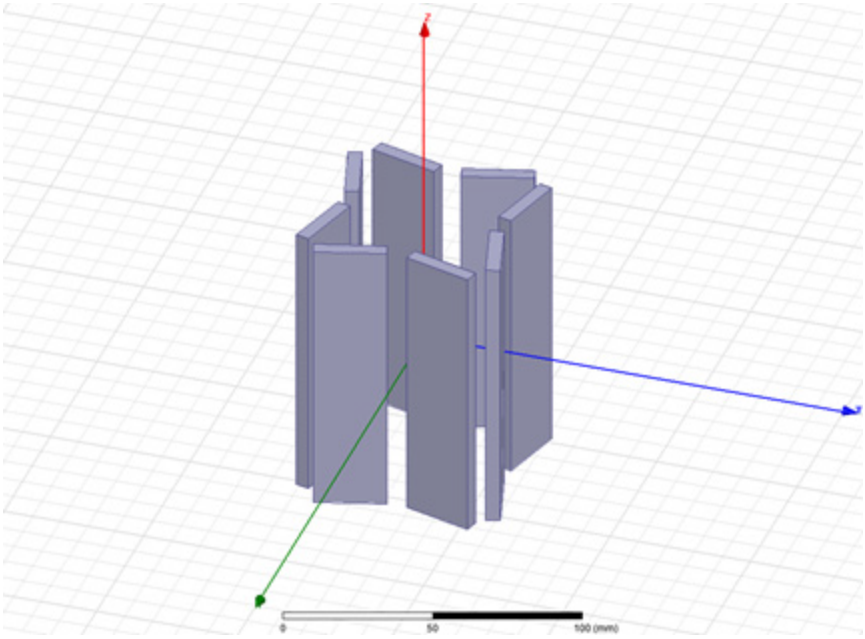


Figure 9-172 All magnets of IPM core

Creating One-Pole Magnets

You can create one-pole magnets of an IPM core manually by setting the property of **InfoCoil** to 3.

You can assign the value of this property either when creating an IPM core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

The following figure shows an example of one-pole magnets.

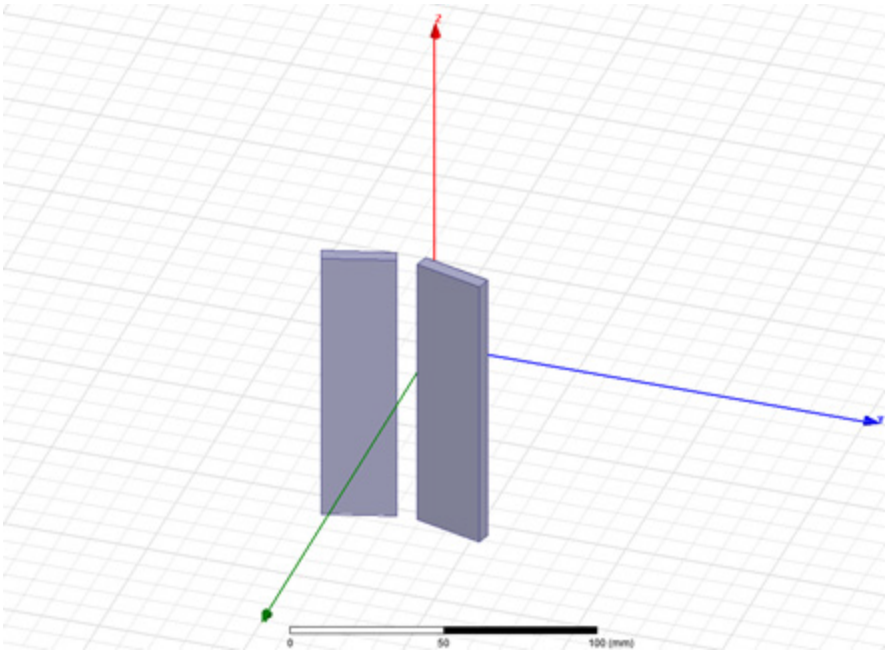


Figure 9-173 One-pole magnets of an IPM Core

PMDamperCore UDP

The PMDamperCore UDP is used to create the PM rotor core with damper for line-start PM synchronous machines.

Property	Description
DiaGap	Core diameter on gap side, or outer diameter.
DiaYoke	Core diameter on yoke side, or inner diameter.
Length	Core length.
Skew	Skew angle in core length range.
Slots	Number of damper slots per pole.

Property	Description
SlotType	Slot type: 1 to 4.
Hs0	Slot opening height.
Hs01	Slot closed bridge height.
Hs1	Slot wedge height.
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.
FilletType	0: a quarter circle; 1: tangent connection; 2&3: arc bottom.
Poles	Number of poles.
PoleType	Pole type: 1 to 8.
D1	Limited diameter of PM ducts.
O1	A locating dimension of PM ducts, for pole types 3, 4, 7 & 8.
O2	A locating dimension of PM ducts, for pole types 1, 3, 5, 6, 7 & 8.
B1	Barrier width, for pole types 1, 2 & 3.
Rib	Rib to hold PM ducts, for pole types 1 to 7.
ThickMag	Magnet thickness.
WidthMag	Total width of all magnet per pole.
BarEndExt	One-side damper bar end extended length.
RingLength	One-side axial ring length.
RingHeight	Radial ring height.
SegAngle	Deviation angle for slot arches (10~30, <10 for true surface).
LenRegion	Region length.
InfoCore	0: core; 1: magnets; 2: damper; 100: region.

These parameters are used in the following figures:

Note	• Parameters O1 and O2 are locating dimensions of PM ducts. They may have different meanings in different pole types. For example, Figure 9-174 shows what O1 and O2 refer to in PMDamper Core with Pole Type 3. • In Figure 9-175 and Figure 9-176 , the gray object (a PMDamperCore with InfoCore set to 0) is used as reference to show the red object (a PMDamperCore with InfoCore set to 2) • For parameters such as **DiaGap, DiaYoke, Length,Skew**, **FilletType**, **SegAngle** and parameters related to slots, refer to SlotCore UDP parameters.

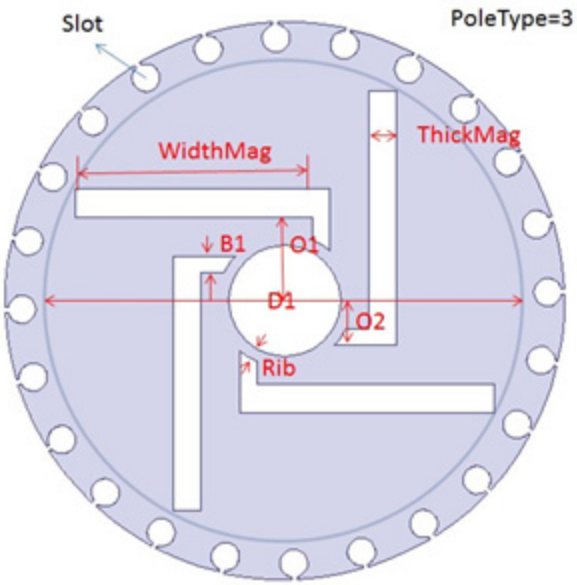


Figure 9-174 PMDamperCore with Pole Type set to 3

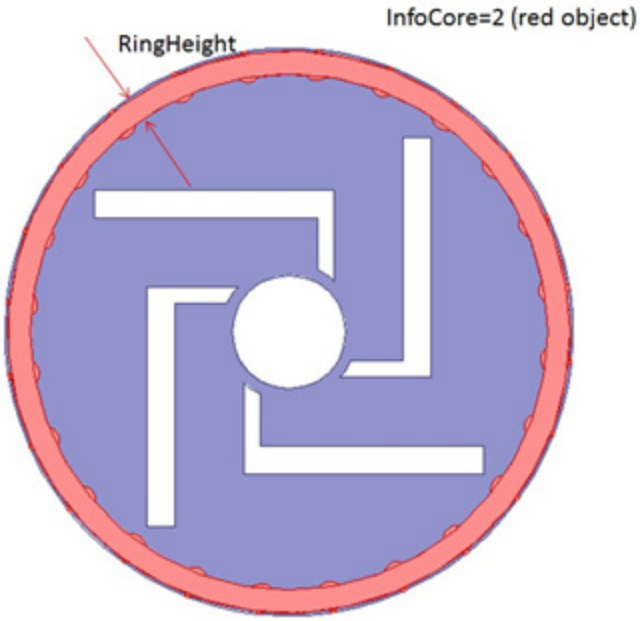


Figure 9-175 Damper of PMDamperCore (red)

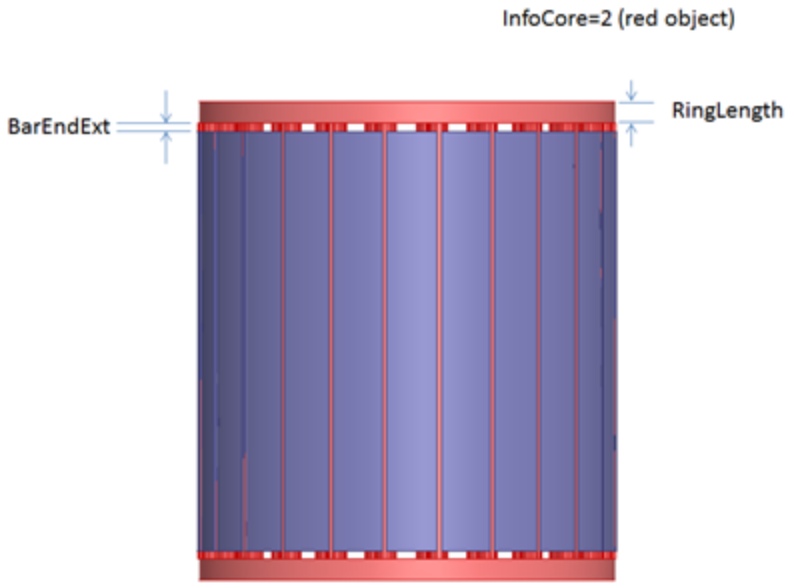


Figure 9-176 3D Damper of PMDamperCore (red)

Creating an Inner PM Damper Core

Set the value of **DiaYoke** and **DiaGap** as **DiaYoke**<**DiaGap** to create an inner PM damper core. You can set the value either when creating the core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Set the **InfoCore** value to 0.

An example is given in the following figure.

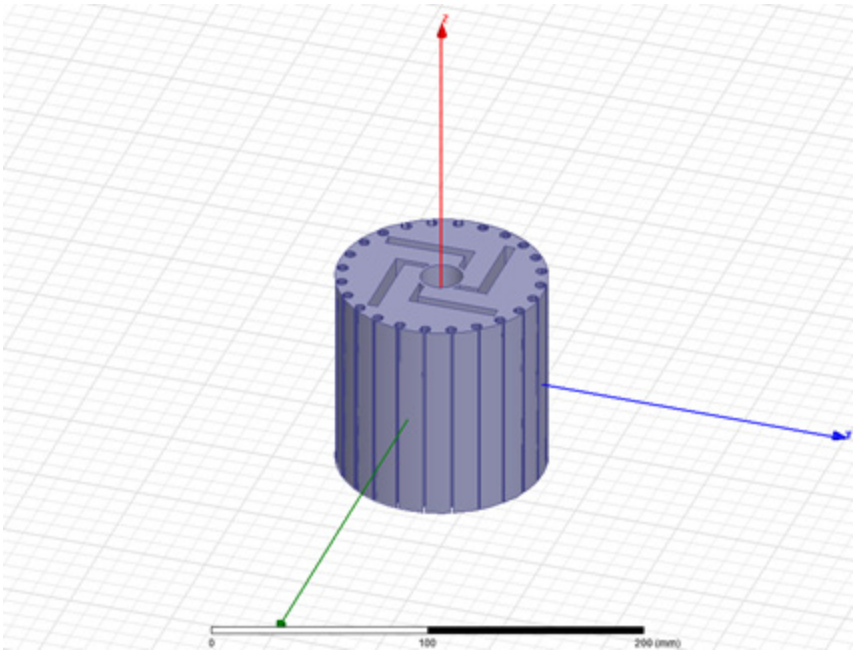


Figure 9-177 Inner PM damper core

Creating a Damper (PM Damper Core)

You create a damper of a PM damper core manually by setting the value of **InfoCoil** to 2.

You can assign the value of this property either when creating a PM damper core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Examples are shown in Figure 9-176 as a red object and Figure 9-178 .

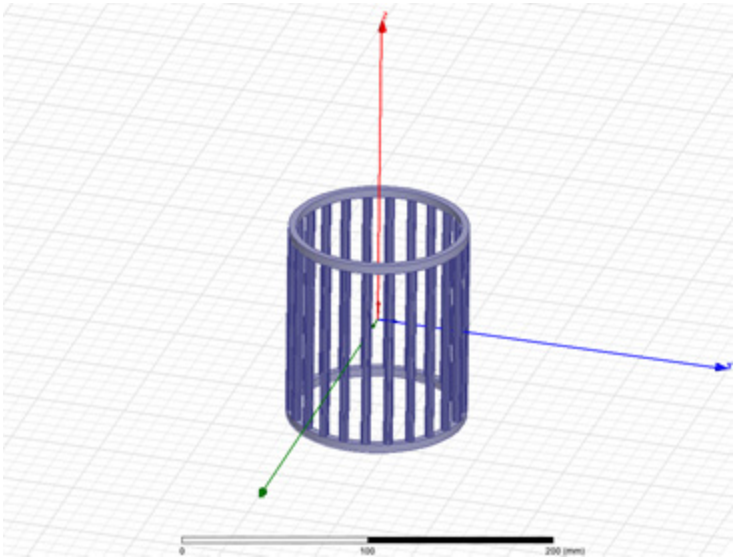


Figure 9-178 A damper

Creating all Magnets (PM Damper Core)

You can create all magnets of a PM damper core manually by setting the value of **InfoCoil** to 1.

You can assign the value of this property either when creating a PM damper core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

An example is shown in the following figure. Note that this example is realized when the value of **PoleType** is 3.

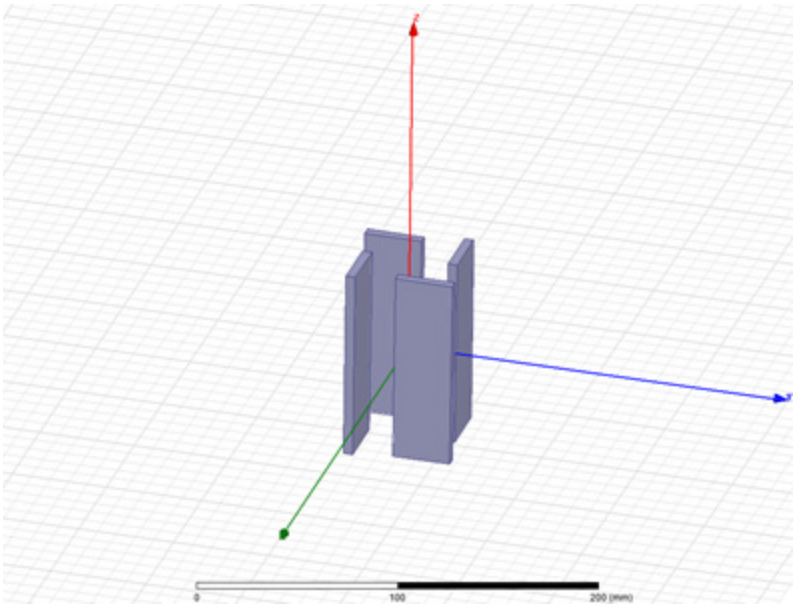


Figure 9-179 All Magnets

Creating a Magnet (PM Damper Core)

You can create a magnet of a PM damper core by setting the **InfoCore** value to -1, -2 or -3 according to the number of magnets for each pole of the core.

For instance, when **Pole Type** is equal to 3 (which is shown in Figure 9-179), there is only 1 magnet for a pole. In this situation, setting the value of **InfoCore** to be -1 can create a magnet. An example is shown in the following figure.

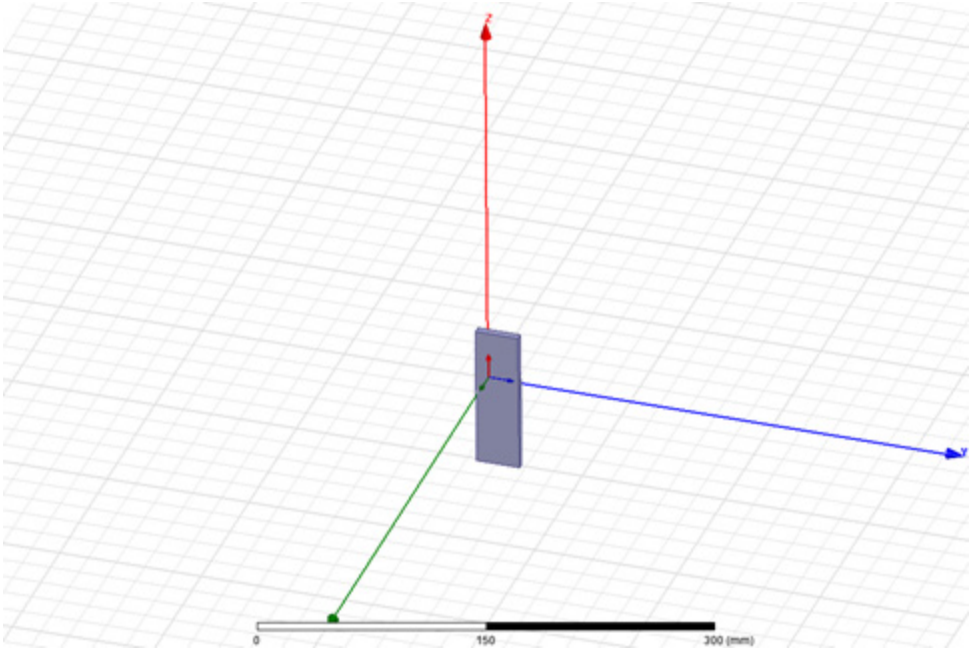


Figure 9-180 A magnet when Pole Type is set to 3

When there are 2 magnets for a pole of a PM damper core (for example, when **Pole Type** is 5), setting the value of **InfoCore** to be -1 can create one of these 2 magnets while setting the value of **InfoCore** to be -2 can create another one.

For instance, Figure 9-181 shows all magnets of a PM damper core when **Pole Type** is 5, whereas Figure 9-182 and Figure 9-183 present one of two magnets for a pole, by setting **InfoCore** value to be -1 and -2 respectively.

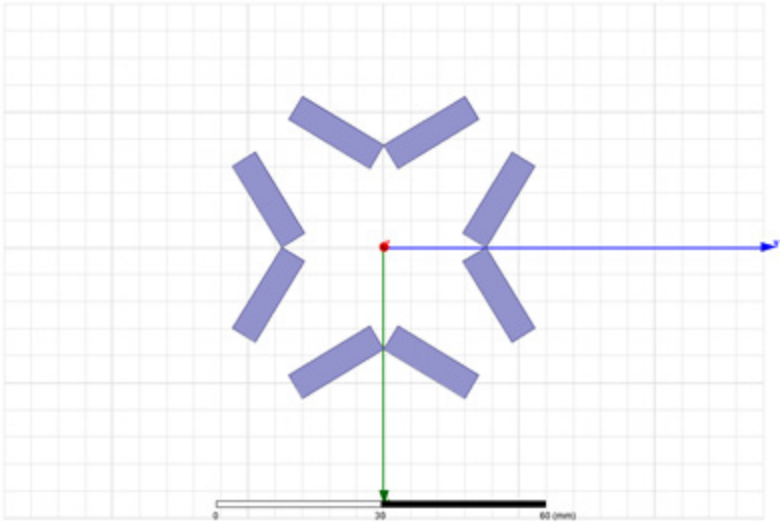


Figure 9-181 A PM damper core when Pole Type is 5 and InfoCore is 1

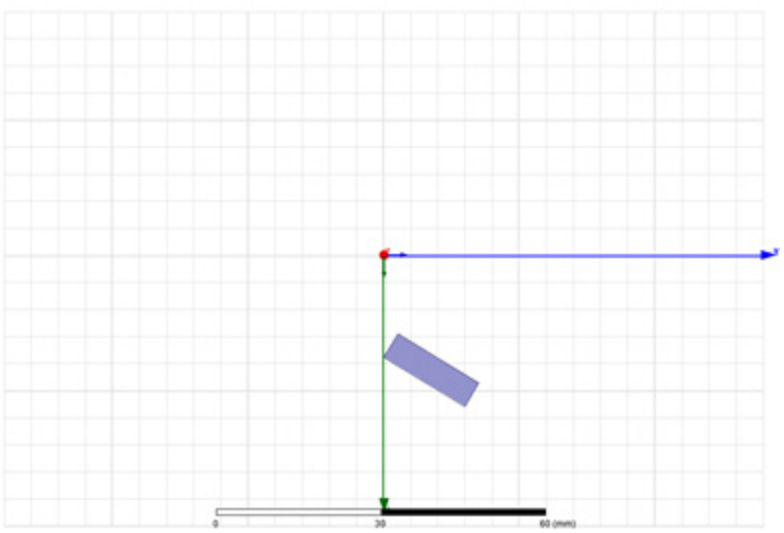


Figure 9-182 A PM damper core when Pole Type is 5 and InfoCore is -1

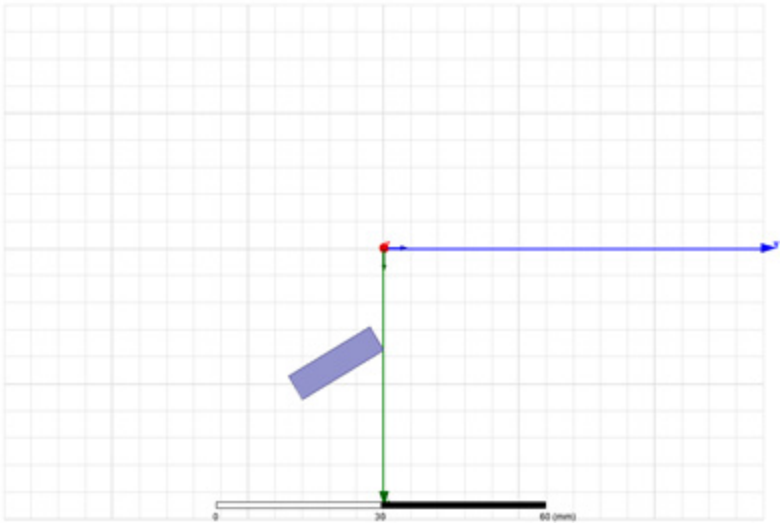


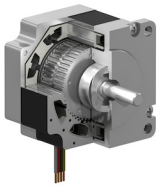
Figure 9-183 A PM damper core when Pole Type is 5 and InfoCore is -2

Similarly, when a pole of PM damper core has 3 or more magnets (such as when **InfoCore** is 6), you can create a single magnet by setting the value of InfoCore to -1, -2, -3 and so on.

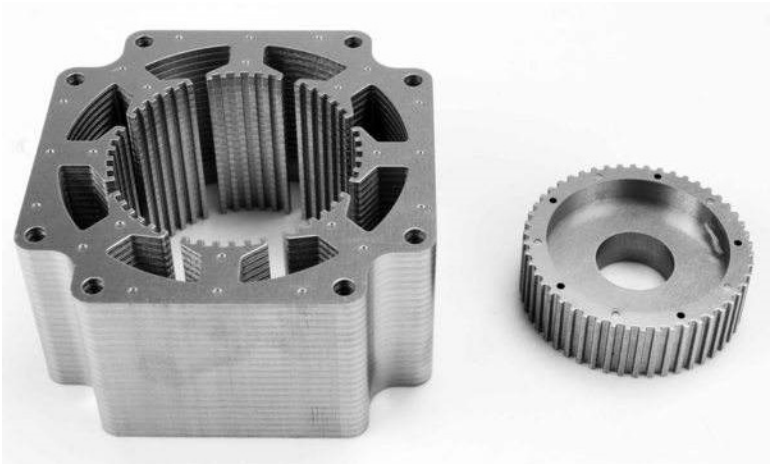
You can assign the value of these properties either when creating a PM damper core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

StepMCore UDP

The StepMCore UDP is used to create stator and rotor cores for Variable Reluctance (VR), or Hybrid, stepper motors, as shown as in Figure 2.3.11.a for a stepper motor, Figure 2.3.11.b for the stator and rotor.



Hybrid Stepper Motor



Stator and rotor of a stepper motor

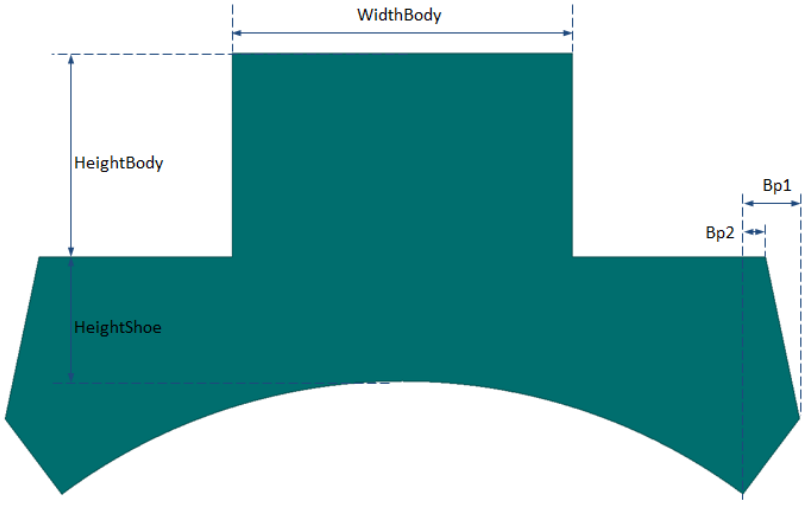
Parameter Definitions

Property	Description
DiaGap	Core diameter on gap side, DiaGap < DiaYoke for outer cores
DiaYoke	Core diameter on yoke side, DiaYoke < DiaGap for inner cores
Length	Core Length
Poles	Number of poles
PoleType	Pole type: 0, 1, or 2 (not valid for inner cores)
EmbraceShoe	Pole shoe embrace related to pole pitch for WidthShoe computation
HeightShoe	Pole shoe height at pole center
WidthBody	Pole body width
HeightBody	Pole body height

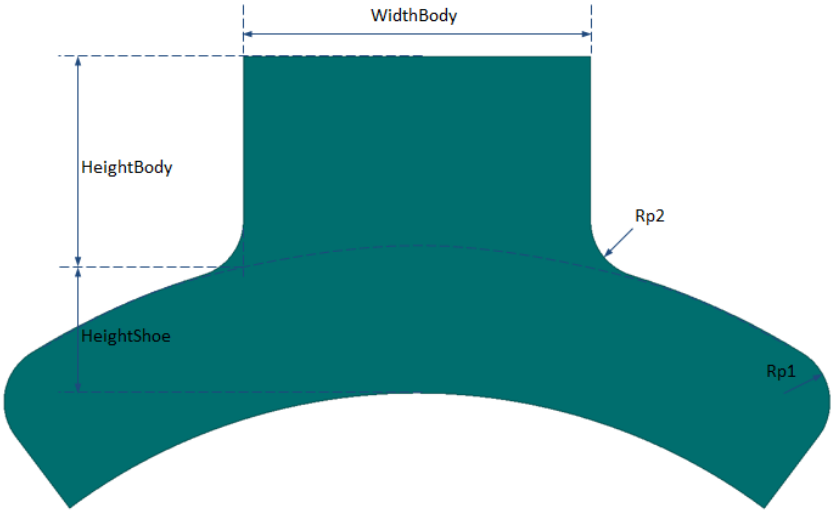
Property	Description
Bp1/Rp1	Bp1 for pole type 1; or Rp1 for pole type 2; not valid for type 0
Bp2/Rp2	Bp2 for pole type 1; or Rp2 for pole type 2; not valid for type 0
SurfaceTeeth	Number of surface teeth per pole (>=1)
IndexingTeeth	Number of indexing teeth for tooth pitch computation, not valid for SurfaceTeeth=1
EmbraceTooth	Tooth embrace related to tooth pitch, not valid for SurfaceTeeth=1
HeightTooth	Surface tooth height, or slot depth, not valid for SurfaceTeeth=1
FilletSlot	Fillet radius at slot bottom
GapCore	Gap of two core sections in radial direction (>0 for Hybrid core)
ThickYoke	Thickness of solid yoke in radial direction for Hybrid core
ThickMag	Magnet thickness in radial direction for Hybrid core
LengthMag	Magnet length in axial direction for Hybrid core (>=GapCore)
CoilEndExt	One-side coil end extended Length
LenRegion	Region length
InfoCore	0: core; 1: core & coils; 2: coil; 3: magnet; 4: terminal1; 5: terminal2; 6: poles; 7: yoke; 100: region

Main parameters described in the table above for different pole types are shown in the figures below.

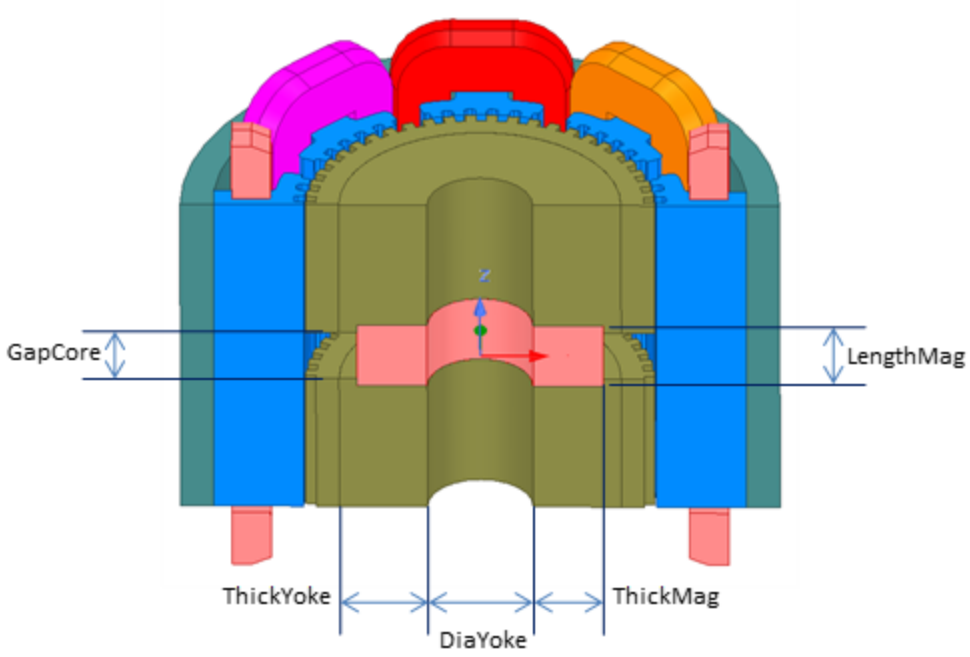
Pole parameters for PoleType = 0



Pole parameters for PoleType = 1



Pole parameters for PoleType = 2



Pole parameters for hybrid rotor

EmbraceShoe is the ratio of the pole shoe angle to pole pitch, and **EmbraceTooth** defines the ratio of a tooth angle to tooth pitch is: ToothPitch = 360°/**IndexingTeeth**.

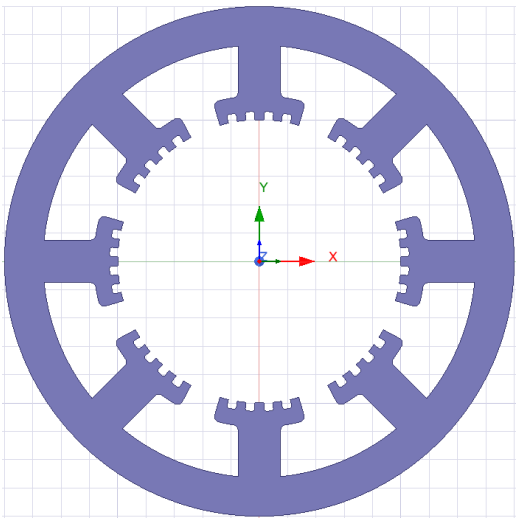
When **SurfaceTeeth** = 1, which means there are no slots on the shoe surface, the shoe width, noted as WidthShoe, will be directly obtained from **EmbraceShoe**. When **SurfaceTeeth** > 1, the minimum shoe angle AngShoeMin will be obtained from the minimum angle to be able to create the specified number of surface teeth, that is

AngShoeMin = (**SurfaceTeeth** – 1 + **EmbraceTooth**) x ToothPitch

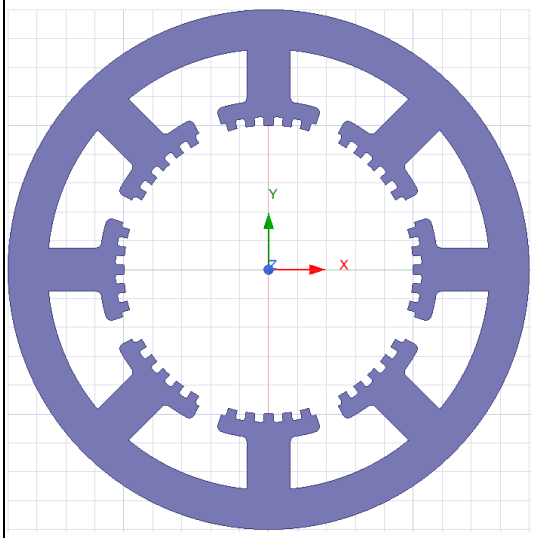
and the maximum shoe angle is defined to add a half slot in each shoe side, that is

AngShoeMax = AngShoeMin + (1 – **EmbraceTooth**) x ToothPitch = **SurfaceTeeth** x ToothPitch.

When the shoe angle corresponding to the specified **EmbraceShoe** is out of the range of (AngShoeMin, AngShoeMax), the UDP will automatically adjust the value of **EmbraceShoe**, as shown below.



EmbraceShoe = 0.7
(Automatically adjust to 0.72)



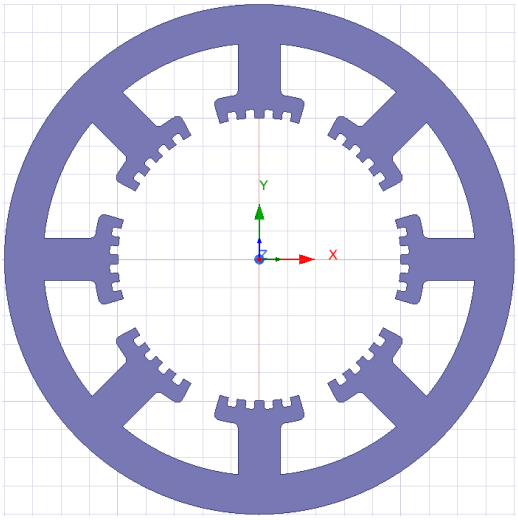
EmbraceShoe = 0.9
(Automatically adjust to 0.8)

In this example, an 8-pole stepper motor stator has **SurfaceTeeth** = 5, **IndexingTeeth** = 50 and **EmbraceTooth** = 0.5. Therefore, ToothPitch = 7.2°, and AngShoeMin and AngShoeMax will be 32.4° and 36°, corresponding to minimum and maximum shoe embrace of 0.72 and 0.8, respectively. The value of **EmbraceShoe** will automatically be adjusted to 0.72, or 0.8, if the input value is out of the range.

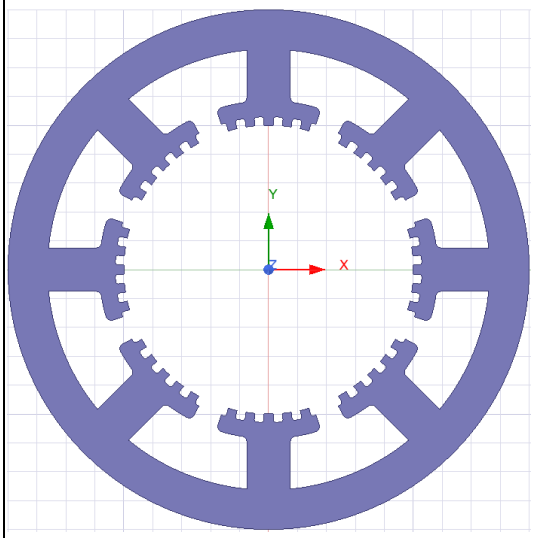
Creating an Inner or Outer Core for a Variable Reluctance Stepper Motor

You can set the value of **DiaYoke** and **DiaGap** as **DiaYoke**<**DiaGap** to create inner cores or **DiaGap**<**DiaYoke** for outer cores.

Keeping **GapCore** = 0 and **InfoCore** = 0, outer stator cores of a variable reluctance stepper motor with different **EmbraceShoe** are created, as shown in the figures below.

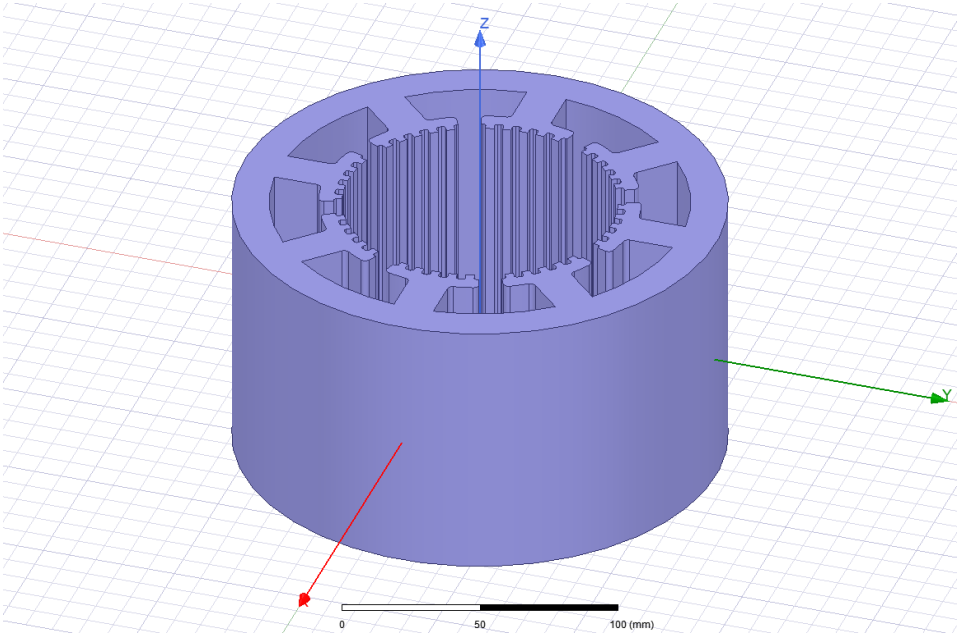


EmbraceShoe = 0.7
(Automatically adjust to 0.72)



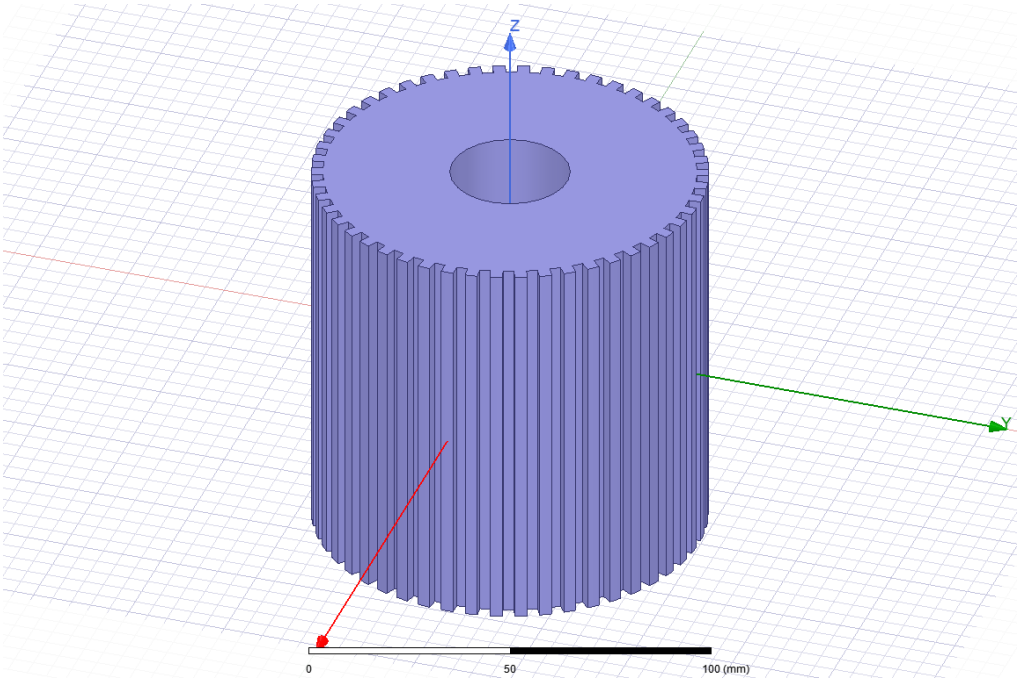
EmbraceShoe = 0.9
(Automatically adjust to 0.8)

Outer stator cores with different shoe embraces in 2D.



Outer stator core in 3D (PoleType = 2).

A 3D inner rotor core with 50 poles, which is the same number as the stator indexing teeth, is shown below



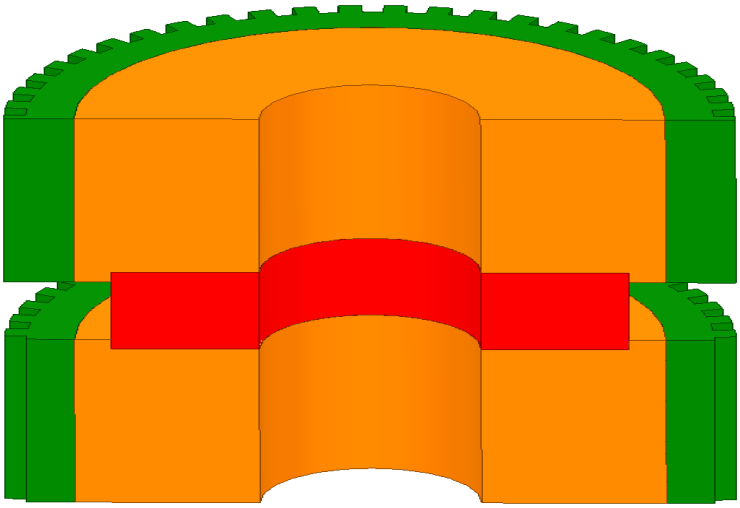
Inner rotor core in 3D (PoleType = 0)

Creating an Inner Rotor Core for a Hybrid Stepper Motor

The stator structure of a hybrid stepper motor is similar to that of a variable reluctance stepper motor, but the rotor structure is quite different. The rotor core is divided into two separate parts. The two parts are shifted by a half tooth pitch in circumferential direction, as shown below. The top pole part, in green color created with **InfoCore** = 6, is cut by the symmetry plan through the pole center, or the tooth center, however, the bottom pole part is cut through the slot center.

The rotor core could be laminated, solid, or laminated for poles and solid for yoke. In the example, the poles, in green color, are laminated, and the yoke, in golden color created with **InfoCore** = 7, is solid. The magnet, in red color created with **InfoCore** = 3, is magnetized in vertical direction. The fluxes in both top and bottom parts are in the radial direction, one from rotor to stator, and the other from stator to rotor.

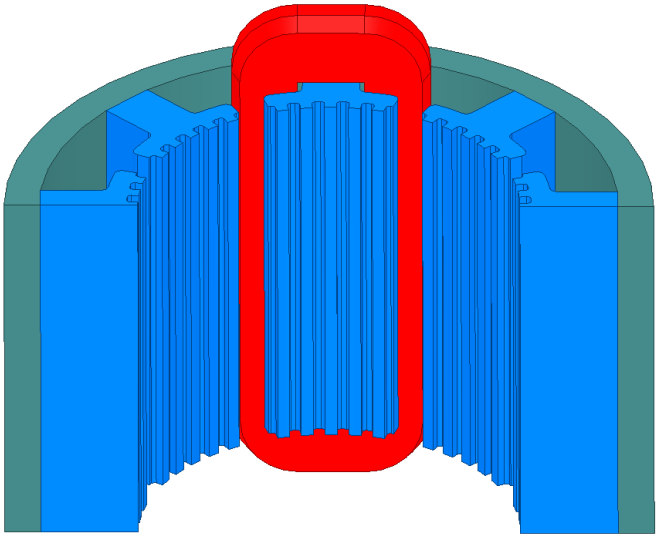
The hybrid rotor can be created only in 3D.



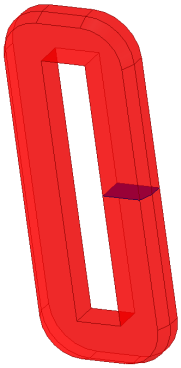
Inner rotor core in 3D (PoleType = 0)

Creating a Coil and Coil Terminals for a Stepper Motor

A coil can be created by setting **InfoCoil** = 2, as shown below,



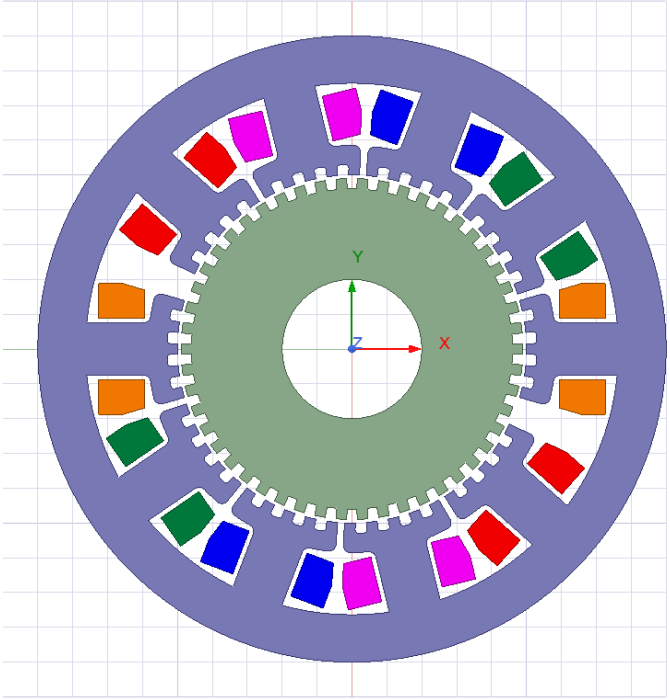
and a coil terminal can be created by setting **InfoCoil** = 4 or 5. (Coil terminal created by **InfoCore** = 4 shown below.)



In 2D, you can create a coil by setting **InfoCoil** = 2 with two united terminals. You can also create two separate terminals by setting **InfoCoil** = 4 and 5 for convenient coil setup.

Creating a Stator Core with Non-uniform Pole Pitch

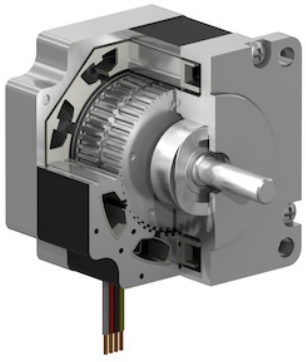
To be able to create torque when stator windings are excited phase by phase (a phase winding normally consists of two coils in two poles shifted by 180 degrees), the number of rotor poles should be the same as the number of indexing teeth for the stator pole surface teeth, and the number of indexing teeth should not be integer multiples of the number of stator poles. However, in some applications, the number of indexing teeth may be integer multiples of the number of stator poles. For example, in widely used 5-phase stepper motors, the stator contains 10 poles, while the rotor has 50 poles, or teeth. The tooth pitch in both stator and rotor is 7.2°. In order to be able to produce continuous torque, the 5 stator pole pairs should be shifted from their uniform positions by 0, 1.44, 2.88, 4.32, 5.76, successively.

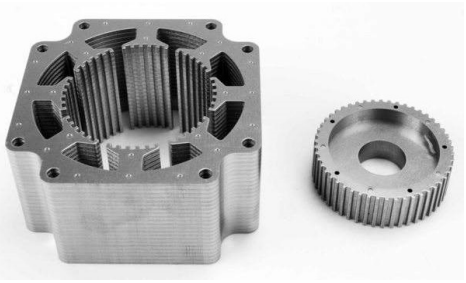


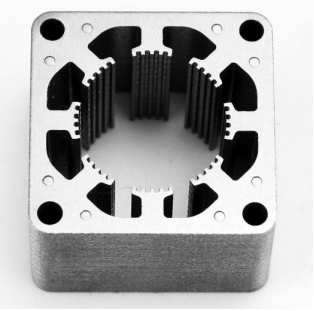
2D Stator core and coils with non-uniform pole pitch

Square Yoke UDP

The SquareYoke UDP is used to create outer square yokes, which can be combined with various stator types to be able to directly mount with end caps for small-size machines, as shown in the examples below.



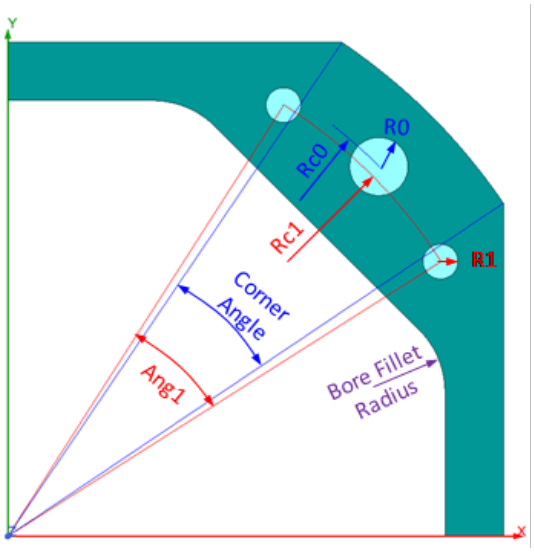




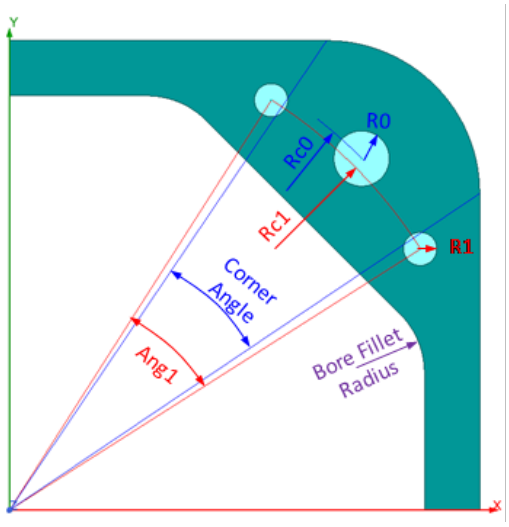
Parameter Definitions

Property	Description
WidOuter	Outer width of square profile
DiaInner	Inner diameter or minimum width
Length	Yoke Length
YokeType	0: Circle; 1: Square Circle; 2: Square Bulged; 3: Square Flat; 4: Square Dented
AngCorner	Corner angle related to bore center, not valid for CornerType = 0
BoreType	1: Circle; 2: Square; 3: Octagon; BoreType is always 1 for CornerType = 0
BoreFillet	Inner bore fillet radius, for square or octagon bore type only
R0	Radius of mounting hole or dented fillet, not valid for CornerType = 0
Rc0	Radius of mounting hole location or bulged fillet, not valid for CornerType = 0
R1	Radius of two corner holes, not valid for CornerType = 0
Rc1	Radius of corner hole location, not valid for CornerType = 0
Ang1	Angle of two corner-hole centers related to bore center, not valid for CornerType = 0
LenRegion	Region length
InfoCore	0: core; 1: holes; 2: 100: region

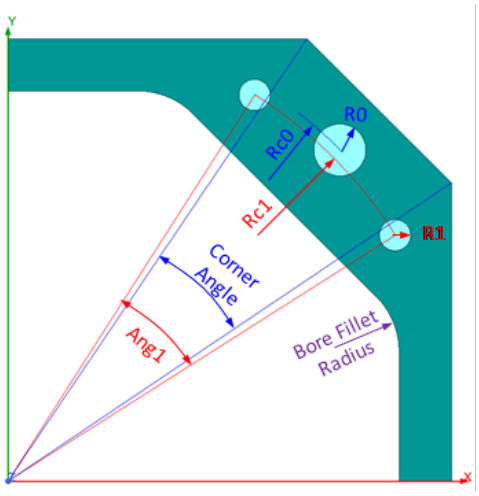
Main parameters described in the table above for the four yoke corner types are shown below, where "Corner Angle" is **AngCorner**, and "Bore Fillet Radius" is **BoreFillet**.



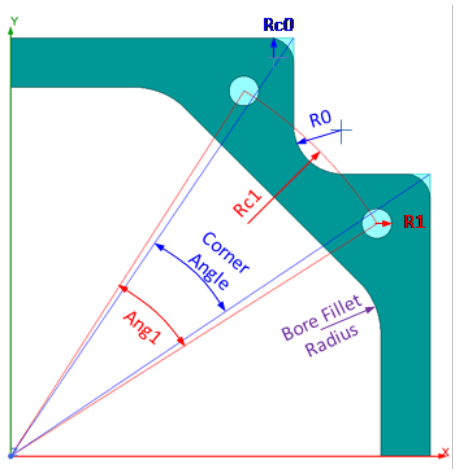
Yoke parameters for YokeType = 1



Yoke parameters for YokeType = 2

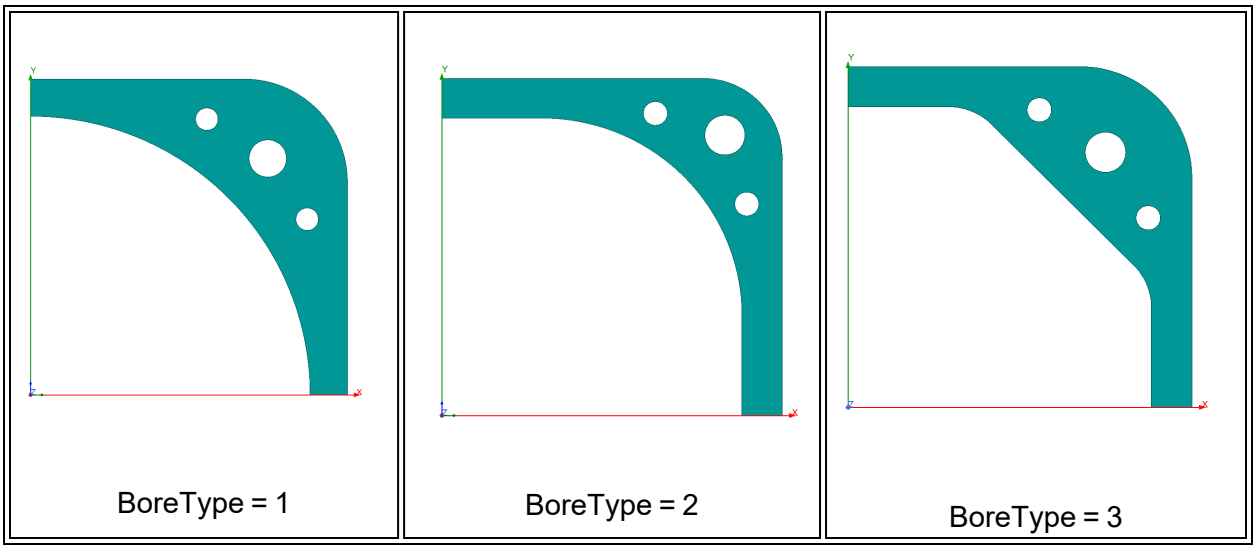


Yoke parameters for YokeType = 3



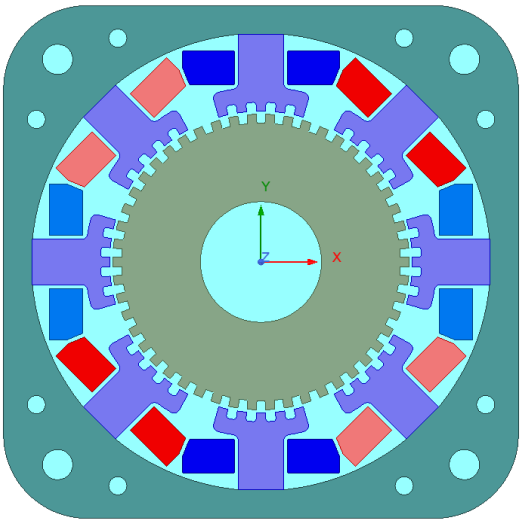
Yoke parameters for YokeType = 4

The three bore types are shown below.

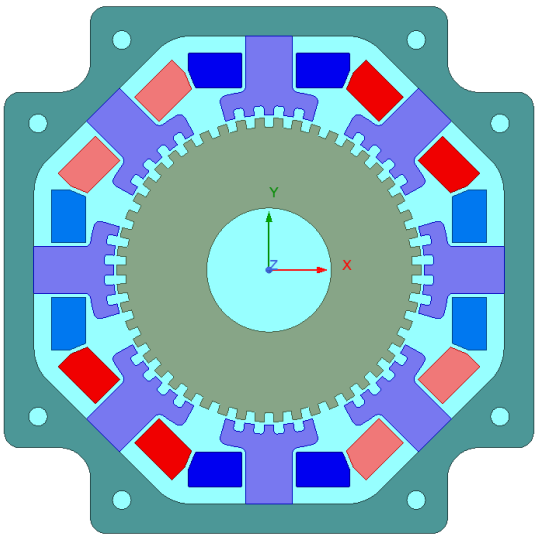


Creating a Stator Yoke to Match Stator Poles of Stepper Motors

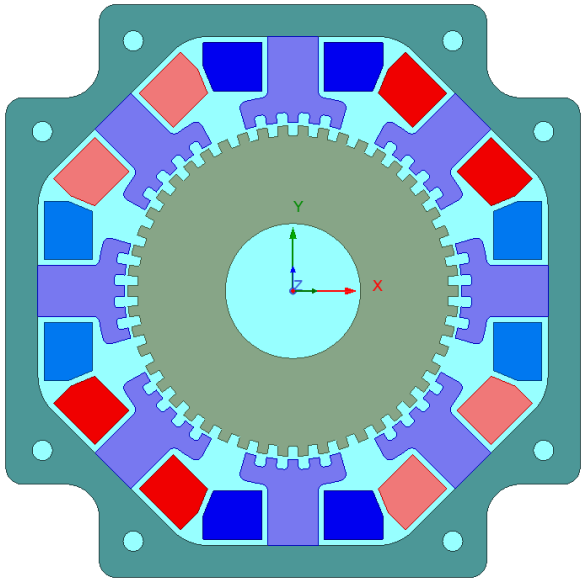
The figure below shows an example for a variable reluctance stepper motor in which the stator poles are mounted to a square yoke with **YokeType**= 2 and **BoreType**= 1, where **BoreFillet** is set as **WidthBody** in StepMCore UDP to fit the pole bottom.



The figure below shows a stepper motor in which the stator poles are mounted to a square yoke with **YokeType**= 4 and octagon **BoreType**= 3 in which the coil area is that of circular bore type (**BoreType**= 1) as shown in the previous figure.



For bore type (**BoreType**= 3), the coil area can be larger. To do this, you can set a negative value for StepMCore UDP parameter **CoilEndExt** to create the coils. The modified coils are shown below.



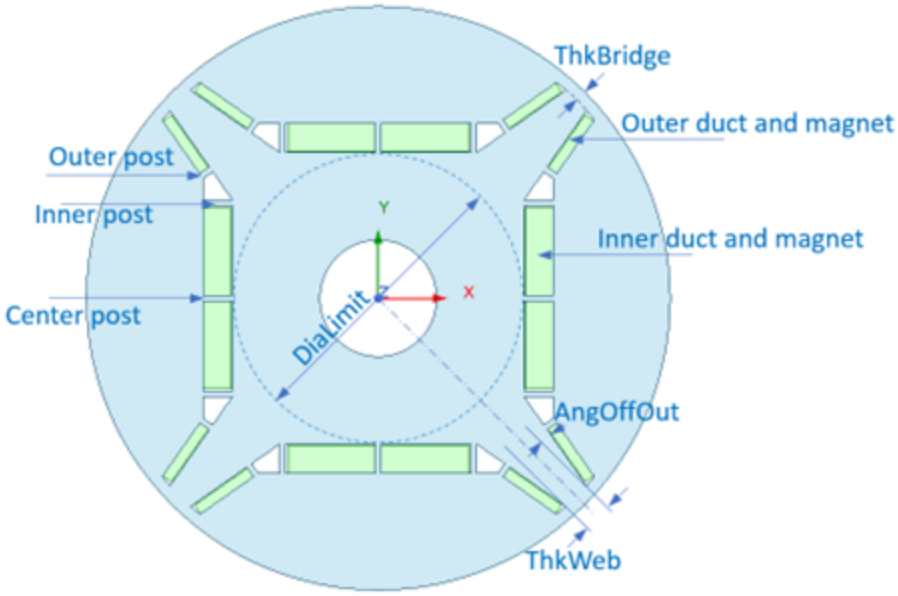
Coil area modified by setting StepMCore UDP parameter **CoilEndExt**< 0.

UPMCore UDP

The UPMCore UDP is used to create a single- or poly-layer U-type IPM core and PMs for IPM synchronous machines, or IPM brushless DC motors.

Property	Description
DiaGap	Core diameter on gap side, or outer diameter.
DiaYoke	Core diameter on yoke side, or inner diameter.
Length	Core length.
Poles	Number of poles.
DiaLimit	Inner limited diameter of PM ducts
ThkBridge	Bridge thickness
ThkWeb	Web thickness
AngOffOut	Outer (web) angle offset
ThkDuctOut	Outer duct thickness
ThkDuctIn	Inner duct thickness
ThkPostOut	Outer post thickness
ThkPostIn	Inner post thickness, <0 for duct bottom corner radius
ThkPostMid	Center post thickness
WidMagOut	One-side width of outer magnets
GapMagOut	Two-side gap of outer magnets (ThkMagOut = ThkDuctOut – GapMagOut)
WidMagIn	Total width of inner magnets per pole
GapMagIn	Two-side gap of inner magnets (ThkMagIn = ThkDuctIn – GapMagIn)
LenRegion	Region length.
InfoCore	0: core; 1: magnets; 2: ducts; 3: inner magnet; 4: left magnet; 5: right magnet; 100: region.

Most parameters for a UPMCore UDP are shown in the figure below except common parameters like **DiaGap**, **DiaYoke**, and **Length**.



Parameters of a UPMCore UDP with **AngOffOut** = -10°.

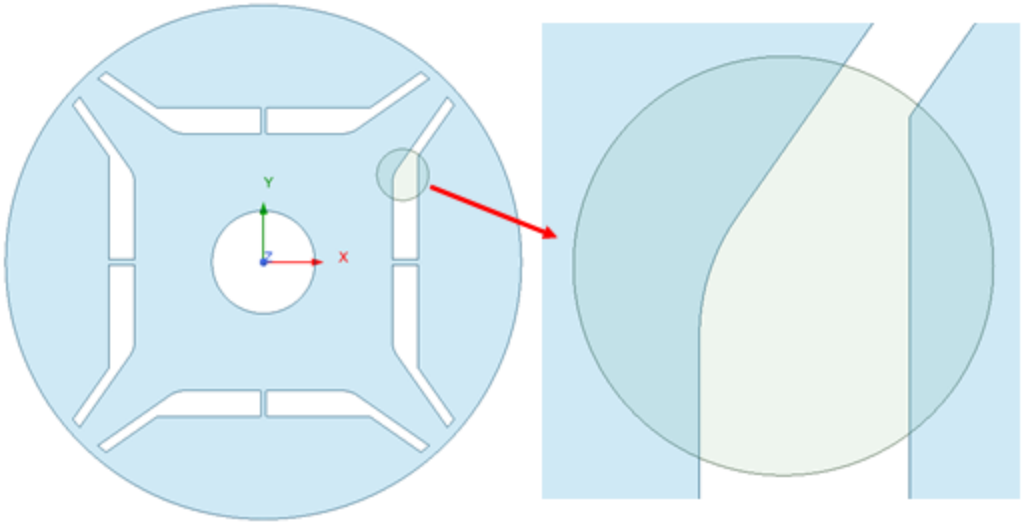
Please notice that when **AngOffOut** = 0, the edges of outer PM ducts between two adjacent poles are parallel. When **AngOffOut** > 0, the inner web thickness is smaller than the outer web thickness which is labeled as **ThkWeb**. When **AngOffOut** < 0, the inner web thickness is larger than the outer web thickness, as shown above.

Creating a U-Type IPM Core

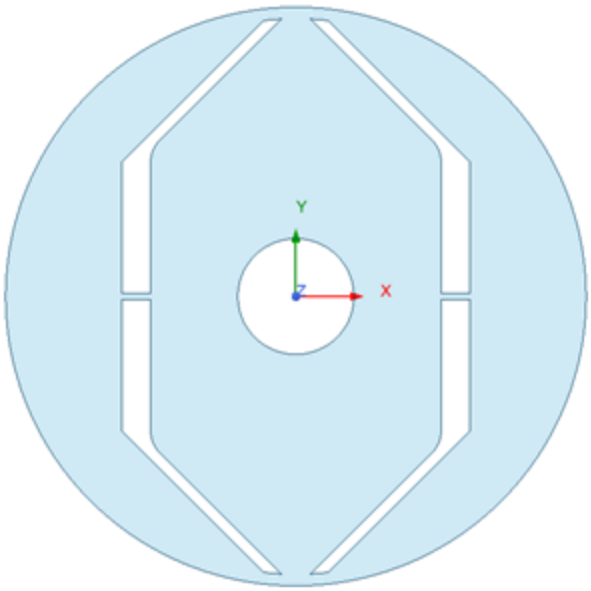
You can create a U-type IPM core with **InfoCore** = 0.

With the default settings of the UPMCore UDP, the created U-type IPM core is shown below.

You can input a negative value for **ThkPostIn**. When **ThkPostIn** < 0, its absolute value, Rc = abs (**ThkPostIn**), is the radius of a fillet at the bottom corner of the PM ducts. The upper limited value of Rc is the maximum value of thickness of the inner and outer ducts. That is, the center point of the fillet arc will always within PM ducts. When **ThkPostIn** < 0, there will be no outer post, that is, the value of **ThkPostOut** will be automatically changed to 0.



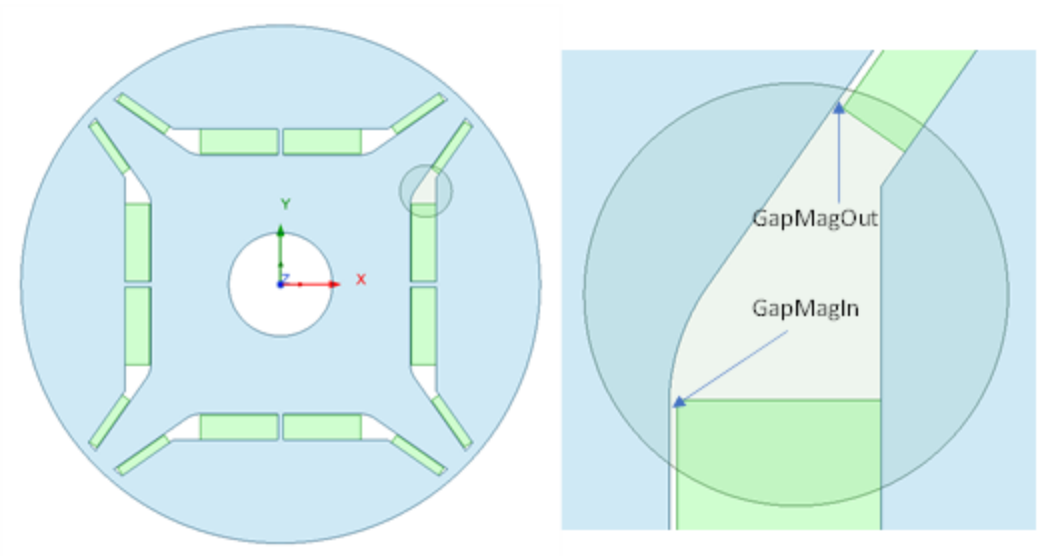
You can also create a U-type IPM core of 2 poles. The following figure is an example of a 2-pole U-type IPM core, where the setting value of **AngOffOut** is -45.



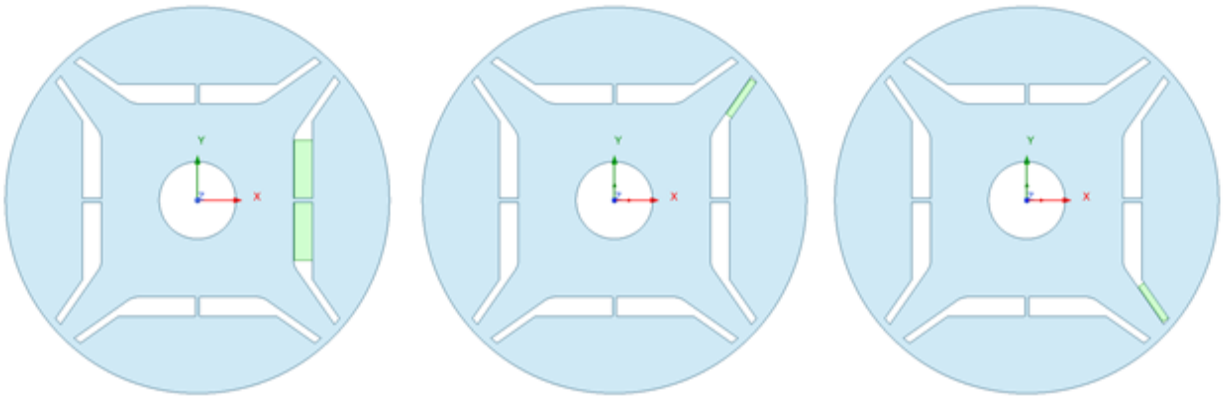
Creating Magnets (UPM Core)

All magnets of a U-type IPM Core are created by setting the value of **InfoCore** to 1, as shown below.

Please notice that **GapMagOut** and **GapMagIn** include all gaps of both top and bottom sides. If both top and bottom gaps are 0.1mm, the created magnets will be located inside the PM ducts in such a way that the top gap is 0, and the bottom gap is 0.2mm. This will not affect the performance solutions. However, this enhances the mesh quality greatly.



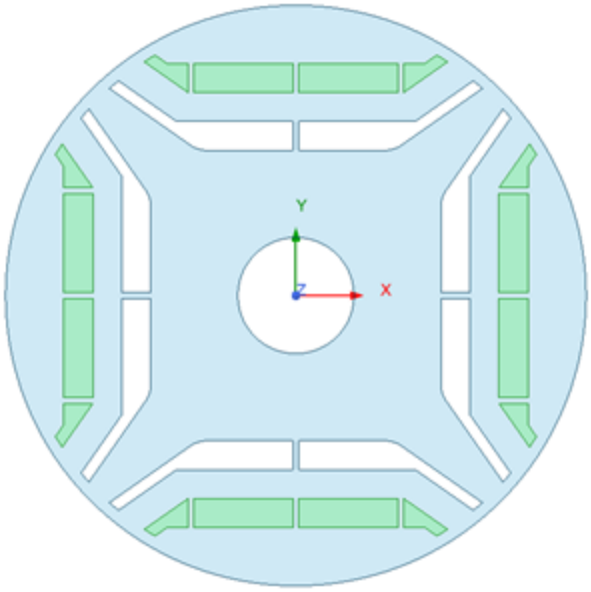
To easily assign the direction of magnet excitation, magnets can be created one-by-one by setting the value of **InfoCore** to 3, 4, or 5. The setting of **InfoCore** = 3 is used to create inner magnet(s) as one object no matter whether it is separated by the center post or not, and the value of **WidMagIn** stands for the total width of inner magnet(s). **InfoCore** = 4 is used to create the left-side magnet, and **InfoCore** = 5 to create the right-side magnet, as shown below.



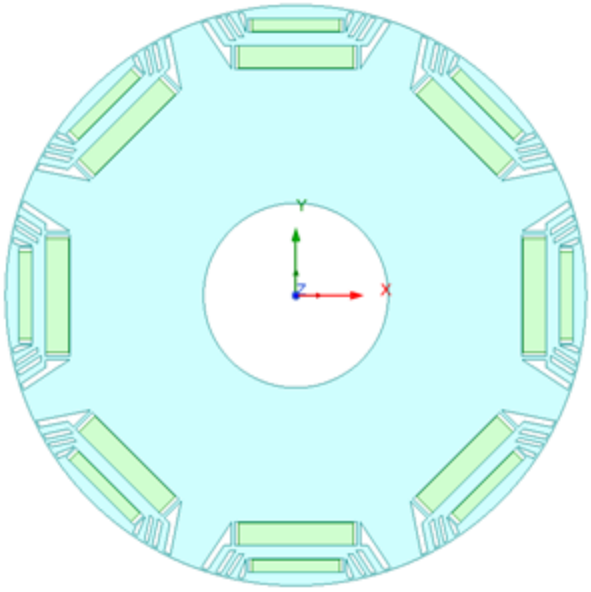
Creating PM Ducts (UPM Core)

All PM ducts of a U-type IPM Core are created by setting the value of **InfoCore** to 2. Subtracting a PM duct set, we can create a poly-layer U-type IPM core

A U-type PM duct set, together with a U-type IPM core, is shown below. The parameters of the additional ducts can be totally different from those of the original core



If you subtract the additional PM duct set from the core, you can get a core with two-layer PM ducts. Continuing to create and subtract additional PM duct sets, we can create an IPM core with an arbitrary number of layers of PM ducts. The figure below shows an IPM core with 4-layer PM ducts. Please notice that when **ThkDuctIn** = 0, the PM ducts will have only the outer parts, and when **ThkDuctOut** = 0, the PM ducts will have only the inner parts.



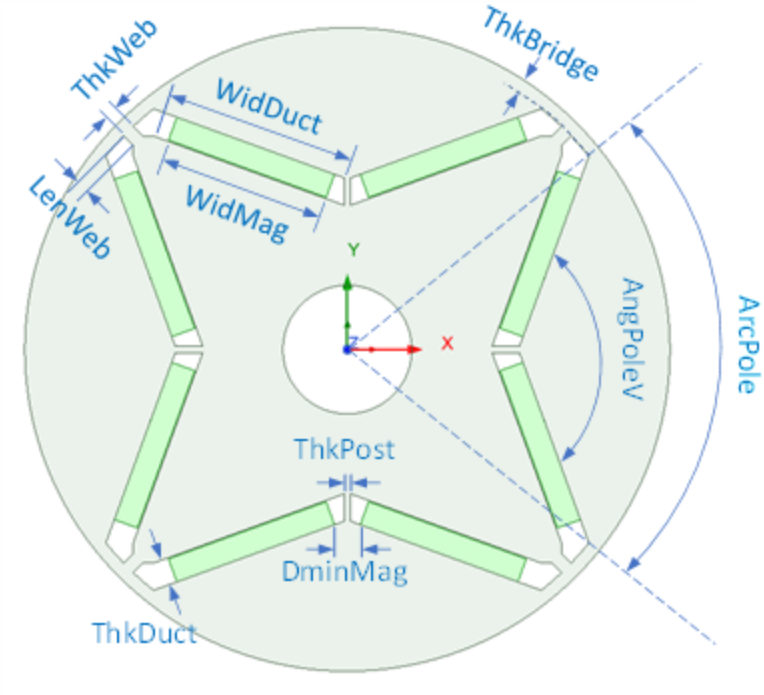
VPMCore UDP

The VPMCore UDP is used to create a single- or poly-layer V-type IPM core and PMs for IPM synchronous machines, or IPM brushless DC motors.

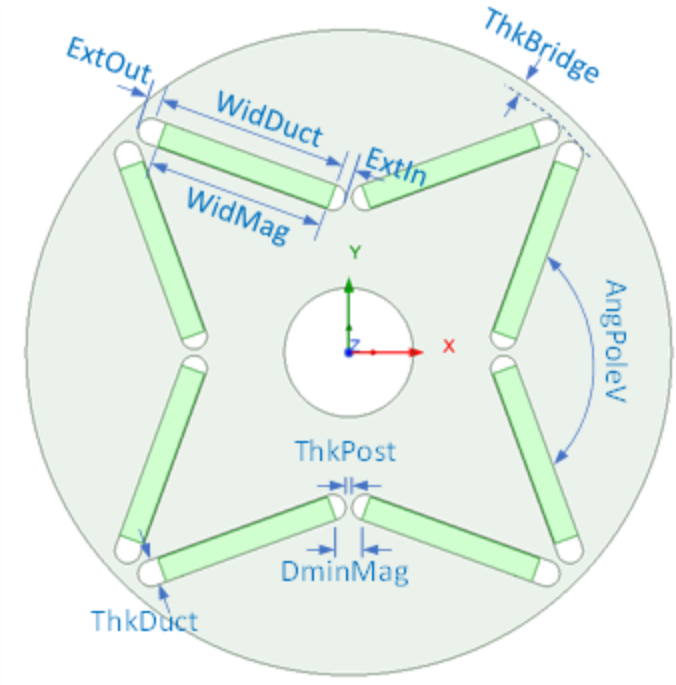
Property	Description
DiaGap	Core diameter on gap side, or outer diameter.
DiaYoke	Core diameter on yoke side, or inner diameter.
Length	Core length.
Poles	Number of poles.
PoleType	0: web; 1: simple.
ThkBridge	Bridge thickness.
ThkDuct	Magnet duct thickness.
WidDuct	Magnet duct width.
ThkWeb/ExtOut	Web thickness (web type), or outer extension (simple type).
LenWeb/ExtIn	Web length (web type), or inner extension (simple type).
AngPoleV	Pole-V angle
ArcPole	Pole arc, in electric degrees, for web pole type only.
ThkPost	Center post thickness, < 0 for duct bottom corner radius (web type)
DminMag	Minimum distance between two-side magnets

Property	Description
WidMag	One-side width of magnets
GapMag	Two-side gap of magnets (ThkMag = ThkDuct – GapMag)
LenRegion	Region length.
InfoCore	0: core; 1: magnets; 2: ducts; 3: left magnet; 4: right magnet; 100: region.

Most parameters for a VPMCore UDP are shown in the figures below except common parameters like **DiaGap**, **DiaYoke**, and **Length**.



Parameters of VPMCore UDP (**PoleType** = 0).



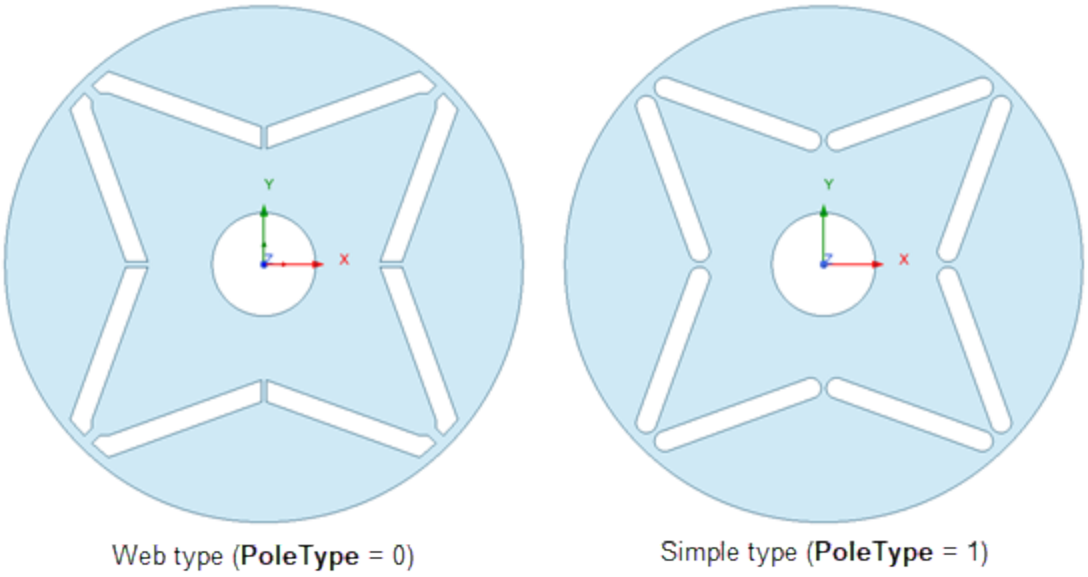
Parameters of VPMCore UDP (**PoleType** = 1).

Creating a V-Type IPM Core

You can create a V-type IPM core by setting **InfoCore** = 0.

With the default settings of the VPMCore UDP, the created V-type IPM cores of web type (**PoleType** = 0) and simple type (**PoleType** =1) are shown below.

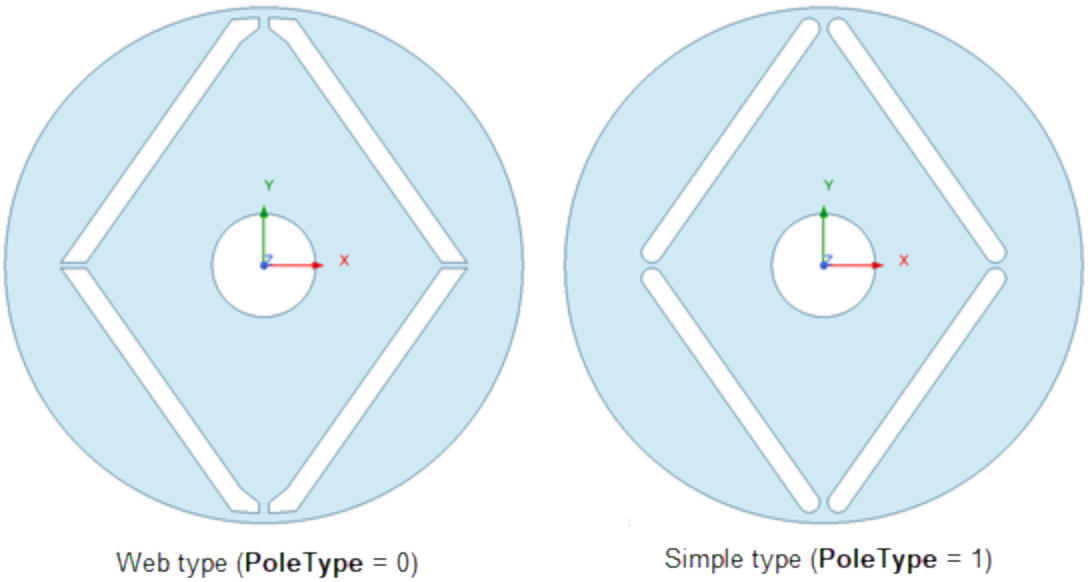
For a web-type core, you can input a negative value for **ThkPost**. When **ThkPost** < 0, its absolute value, Rc = abs(**ThkPost**), is the radius of a fillet at the bottom corner of the PM ducts. The upper limited value of Rc is the thickness of the PM ducts. For a simple-type core, the upper limit value for **ExtOut** and **ExtIn** is a half thickness of PM ducts.



Web type (**PoleType** = 0) Simple type (**PoleType** = 1)

4-pole V-type IPM cores

We can create a V-type IPM core of 2 poles. The figure below is an example of 2-pole V-type IPM core, where the value of **AngPoleV** is 250.



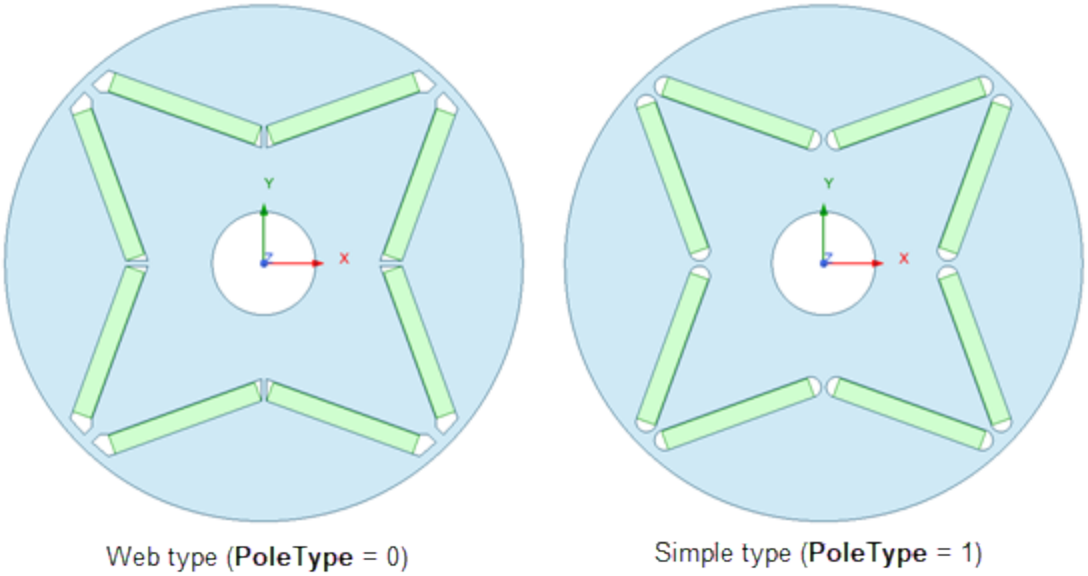
Web type (**PoleType** = 0) Simple type (**PoleType** = 1)

2-pole V-type IPM cores

Creating Magnets (VPM Core)

All magnets of a V-type IPM Core are created by setting the value of **InfoCore** to 1. V-type IPM cores of web type and simple type with all magnets are shown below.

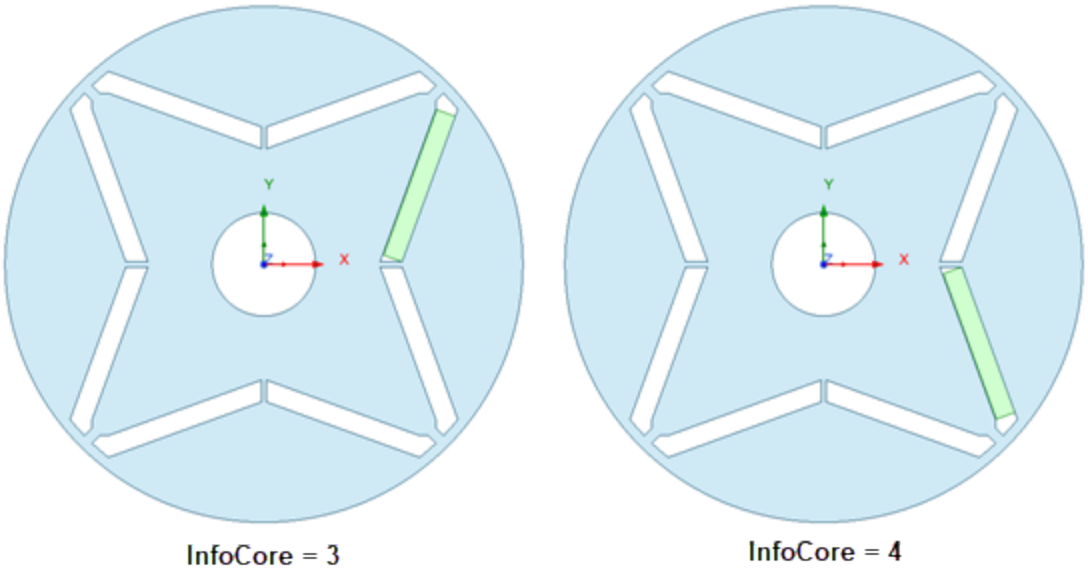
Please notice that **GapMag** includes total gaps of both top and bottom sides. If both top and bottom gaps are 0.1mm, the created magnets will be located inside the PM ducts in such a way that the top gap is 0, and the bottom gap is 0.2mm. This will not affect the performance solutions. However, this enhances the mesh quality greatly.



Web type (**PoleType** = 0) Simple type (**PoleType** = 1)

V-type IPM cores with all magnets

To easily assign the direction of magnet excitation, magnets can be created one-by-one by setting the value of **InfoCore** to 3, or 4. The setting of **InfoCore** = 3 is used to create the left-side magnet and **InfoCore** = 4 is used to create the right-side magnet as shown below.



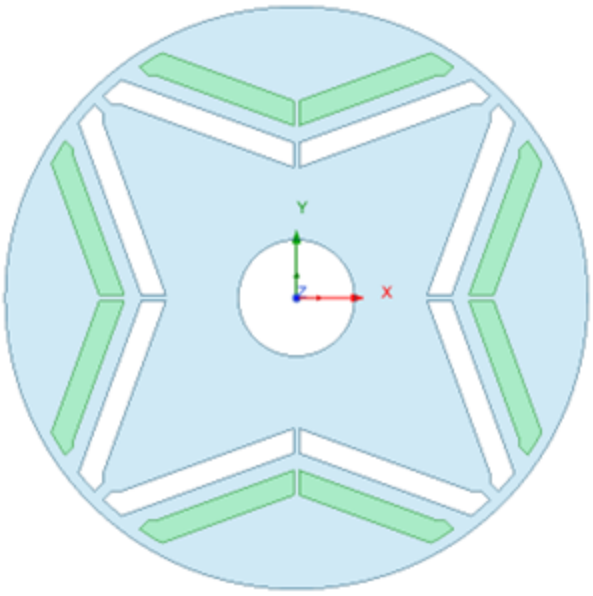
V-type IPM core with a single magnet

When the input values of magnet width are too large to fit in the PM ducts, they will be modified based on the maximum possible sizes.

Creating PM Ducts (VPM Core)

All PM ducts of a V-type IPM Core are created by setting the value of **InfoCore** to 2. Subtracting a PM duct set, you can create a poly-layer V-type IPM core

A V-type PM duct set, together with a V-type IPM core, is shown below. The parameters of the additional ducts can be totally different from those of the original core.



If you subtract the additional PM duct set from the core, you can get a core with two-layer PM ducts. Continuing to create and subtract additional PM duct sets, we can create an IPM core with an arbitrary number of layers of PM ducts.

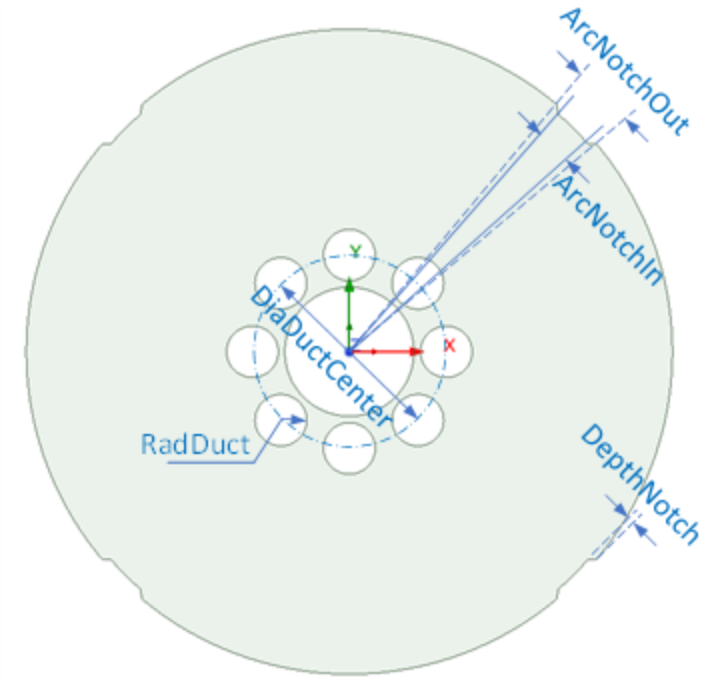
VentCore UDP

The VentCore UDP is used to create a vent core, a vent shaft, vent ducts, as well as notches, for various cores of electrical machines. A vent core can be used as a base object from which some tool objects can be subtracted to form various machine cores. The vent ducts and notches can also be used as tools to be subtracted from other machine cores.

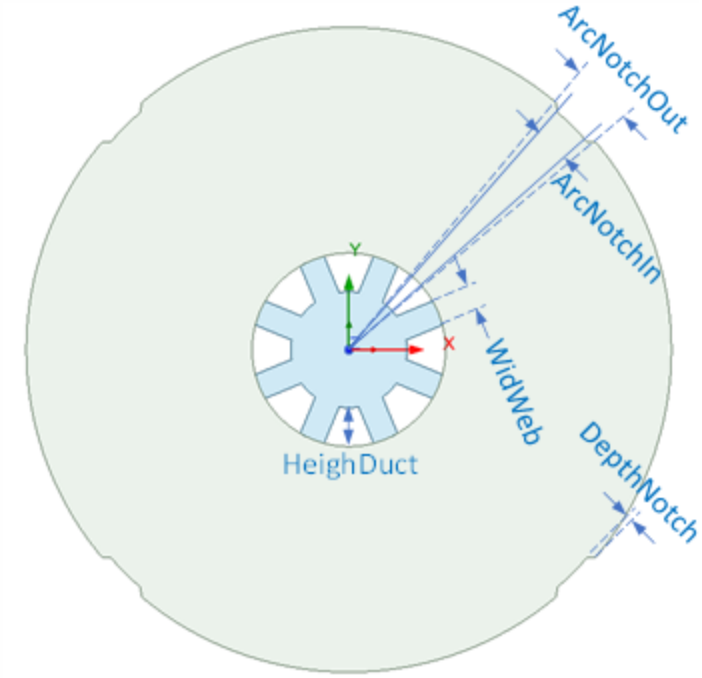
Property	Description
DiaGap	Core diameter on gap side, or outer diameter.
DiaYoke	Core diameter on yoke side, or inner diameter.
Length	Core length.
Poles	Number of poles for notches.
DepthNotch	Notch depth.
ThkBridge	Bridge thickness.
ArcNotchOut	Notch outer arc, in electric degrees.
ArcNotchIn	Notch inner arc, in electric degrees.
Ducts	Number of axial vent ducts.
DuctType	0: none; 1: circular; 2: shaft spoke; 3: arc; 4: rectangular.

Property	Description
DiaDuctCenter	Diameter of duct center (not for shaft spoke).
HeightDuct	Duct height in radial direction.
WidDuct/WidWeb	Duct width (rectangular) or web width (shaft spoke, arc).
RadDuct	Duct radius (circular) or corner radius (arc, rectangular).
AngDuctShift	Duct shift angle.
LenRegion	Region length.
InfoCore	0: core; 1: shaft; 2: ducts; 3: notches; 100: region.

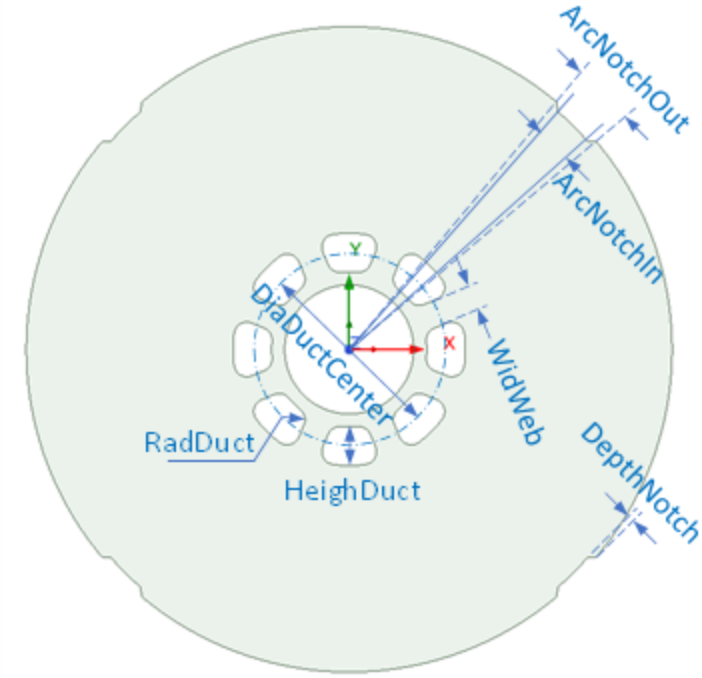
Most parameters for a VentCore UDP are shown in the figures below except common parameters like **DiaGap**, **DiaYoke**, and **Length**.



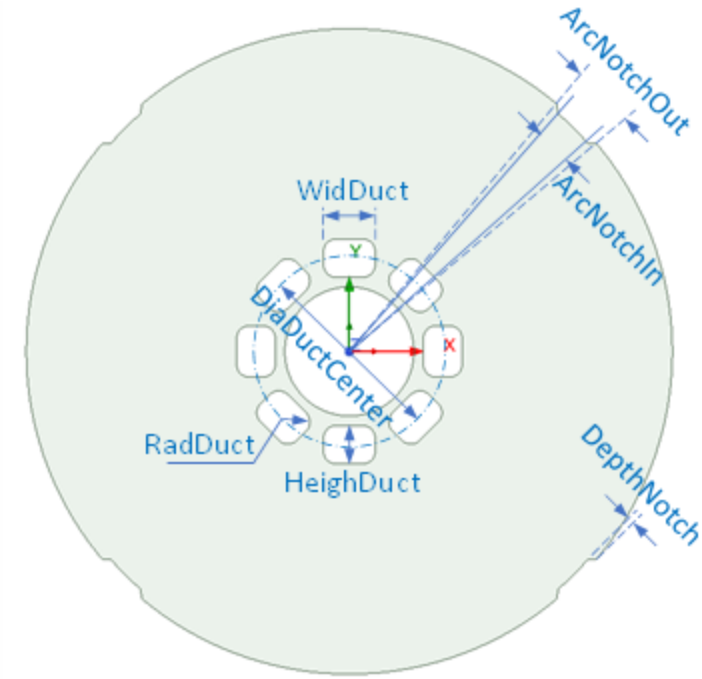
Parameters of VentCore UDP (**DuctType** = 1).



Parameters of VentCore UDP (**DuctType** = 2).



Parameters of VentCore UDP (**DuctType** = 3).

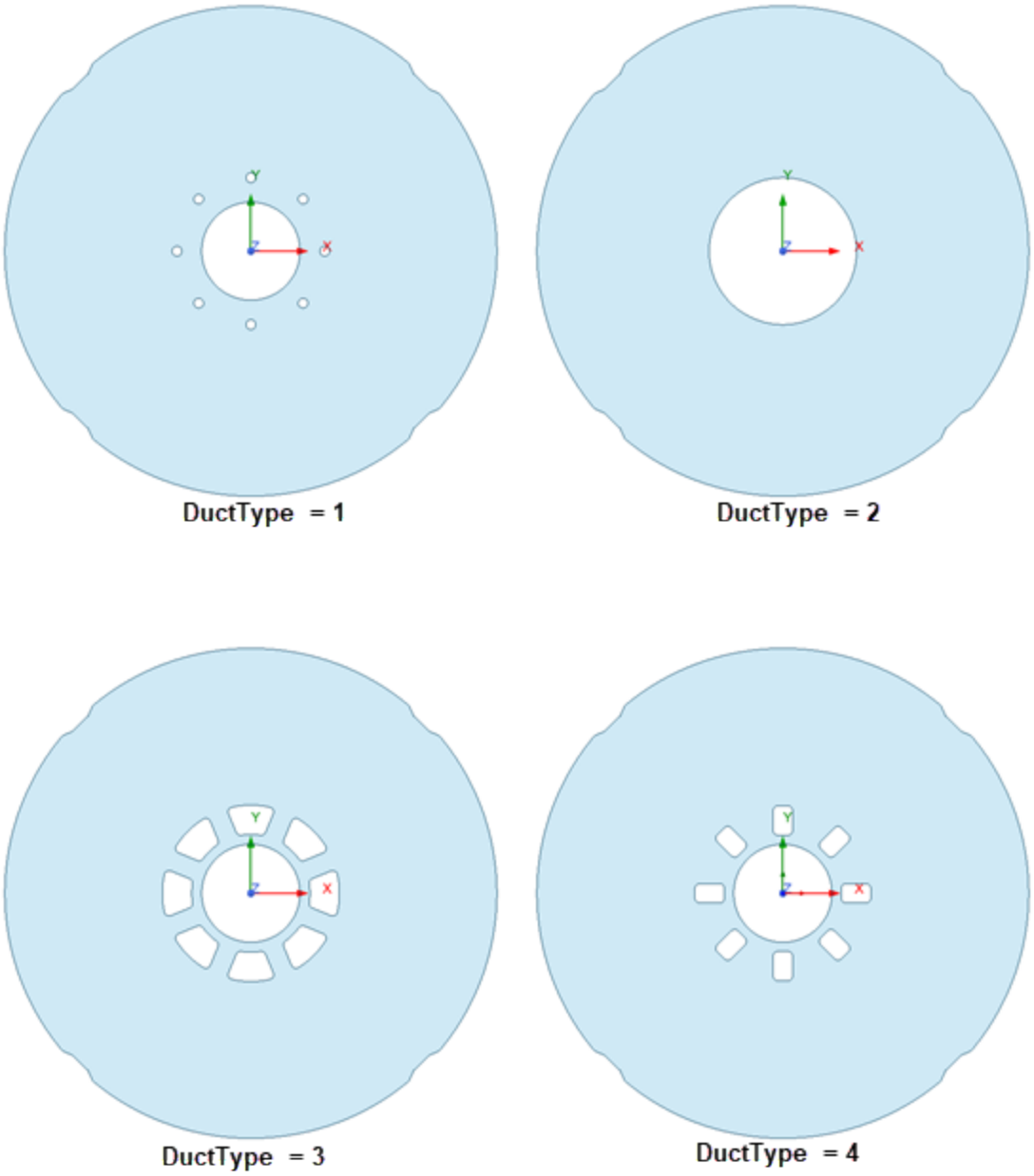


Parameters of VentCore UDP (**DuctType** = 4).

Creating a VentCore

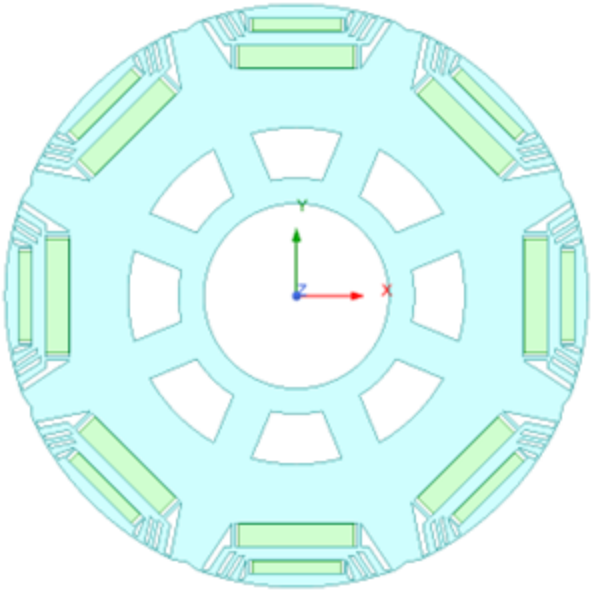
You can create a vent core from VentCore UDP by setting **InfoCore** = 0.

With the default parameter settings of the VentCore UDP, the created vent cores of different duct types are shown in the figures below. Note that the value of **DiaYoke** for **DuctType** = 2 is modified to 30mm to pass the validation.



VentCores of various Duct Types

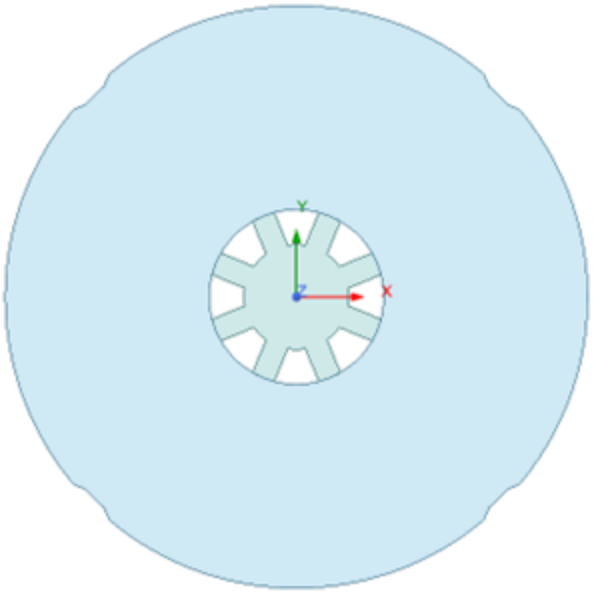
The following figure shows an example in which a vent core is used as a base object, and several U-type IPM ducts are used as tool objects to be subtracted from the vent core to form a poly-layer IPM core with vent ducts.



A poly-layer IPM core with vent ducts.

Creating a Shaft (VentCore)

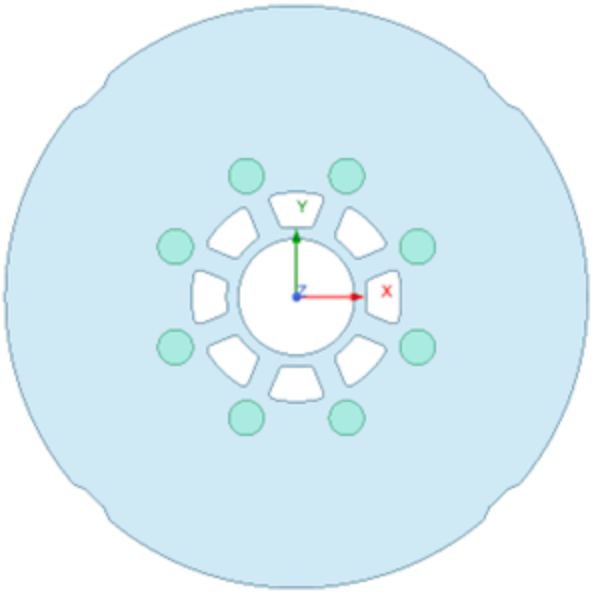
You can create a vent core shaft using the VentCore UDP by setting the value of **InfoCore** to 1. A shaft, together with a V-type IPM core with **DuctType** = 2, is shown below.



Creating Vent Ducts (VentCore)

All vent ducts of a vent core are created by setting the value of **InfoCore** to 2. Subtracting a vent duct set, you can create a vent core with poly-layer vent ducts.

A vent duct set, together with a vent core, is shown in the figure below. The parameters of the additional ducts can be totally different from those of the original core.



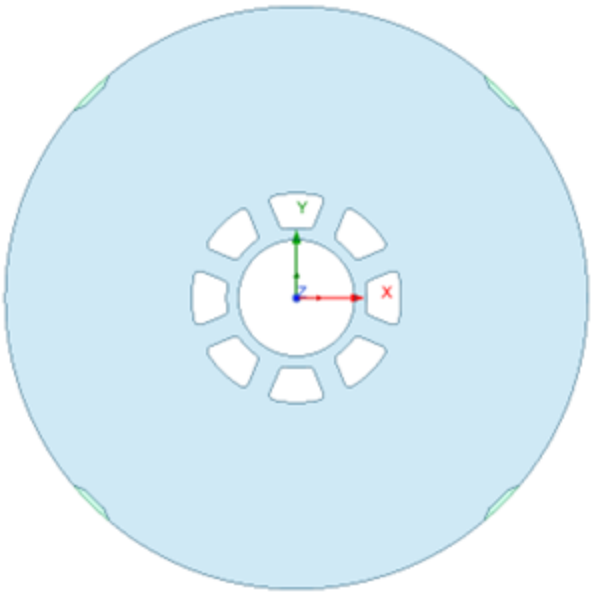
A vent duct set of **DuctType** = 1, with a vent core of **DuctType** = 3.

If you subtract the additional vent duct set from a vent core, you can get a core with two-layer vent ducts. Continuing to create and subtract additional vent duct sets, you can create a vent core with an arbitrary number of layers of vent ducts.

Creating Notches (VentCore)

All notches of a vent core are created by setting the value of **InfoCore** to 3, which can be subtracted from an IPM core to improve the air-gap flux density waveforms.

All created notches, together with a vent core of **DuctType** = 3, are shown in the figure below. The parameters of the additional Notches can be totally different from those of the original core.



Notches together with a vent core of **DuctType** = 3.

UDPs for Other Cores and Coils

UDPs defined for other cores and coils include:

- DiskSlotCore UDP
- DiskCoil UDP
- DiskPMCore UDP
- LinearMCore UDP
- TransCore UDP
- TransCoil UDP
- TransTwist UDP

DiskSlotCore UDP

The DiskSlotCore UDP is used to create a stator/rotor core for axial-flux machines. It can also create a squirrel-cage winding of the axial-flux type.

Property	Description
DiaOuter	Core outer diameter.
DiaInner	Core inner diameter.
Thickness	Core axial thickness per side.
Gap	Gap between core & xy plane (>0: upper; <0: lower; =0: double-sided).

Property	Description
Skew	Skew angle.
Slots	Number of slots.
SlotType	Slot type: 1 to 6.
Hs0	Slot opening height.
Hs01	Slot closed bridge height.
Hs1	Slot wedge height.
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.
FilletType	0: a quarter circle; 1: tangent connection.
HalfSlot	0: symmetric slots; 1: half slots.
RingLength	One-side radial ring length.
RingHeight	Axial ring height.
SegAngle	Deviation angle for slot arches (10~30, <10 for true surface).
InfoCore	0: core only; 1: squirrel cage only.

These parameters are used in the following figures.

Note	
	• For parameters such as **Skew**, **FilletType**, **SegAngle**, **HalfSlot** and parameters related to slots, refer to SlotCore UDP parameters. • Figures are shown in pairs to compare parameters when the **InfoCore** value varies.

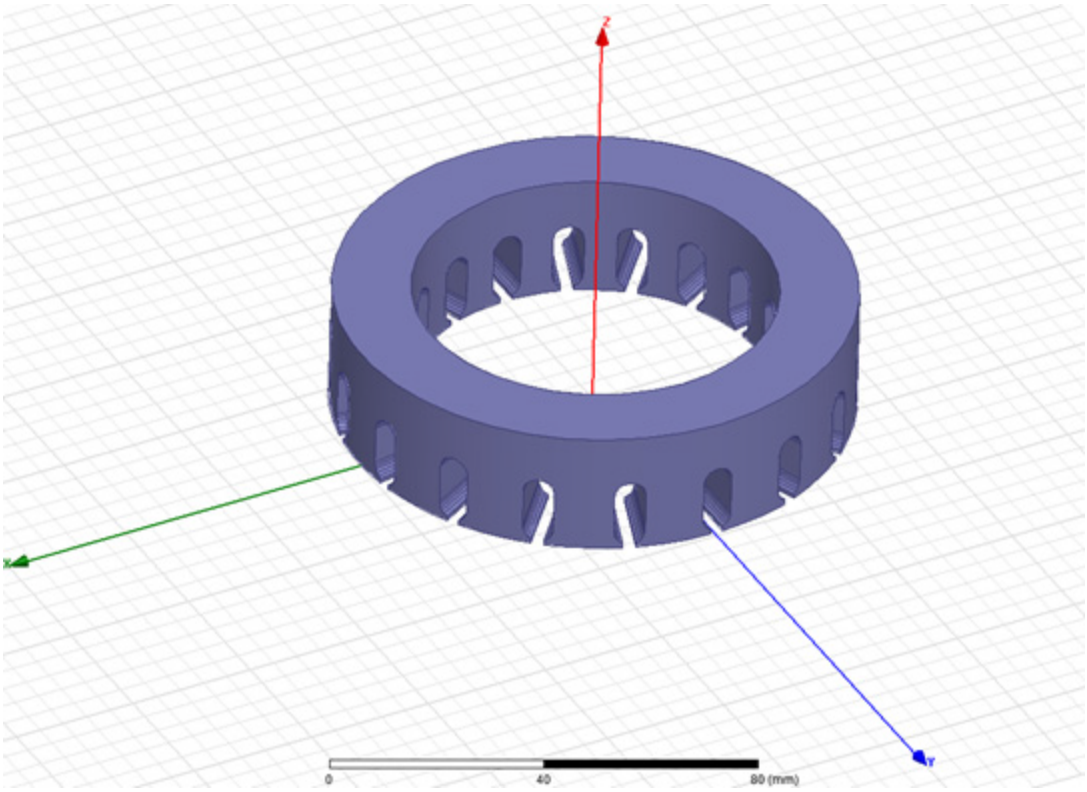


Figure 9-184 Overview of DiskSlotCore (InfoCore set to 0)

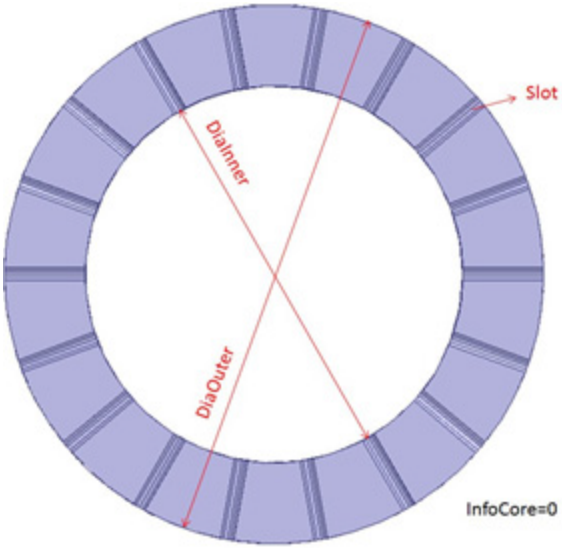


Figure 9-185 DiskSlot Core (InfoCore set to 0)

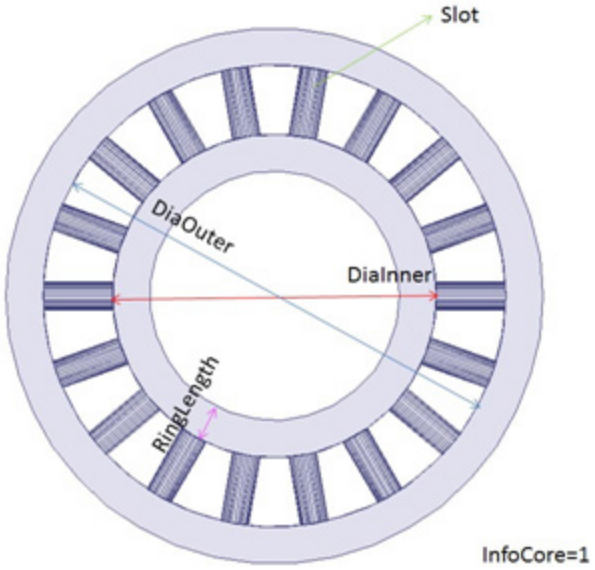


Figure 9-186 DiskSlot Core (InfoCore set to 1)

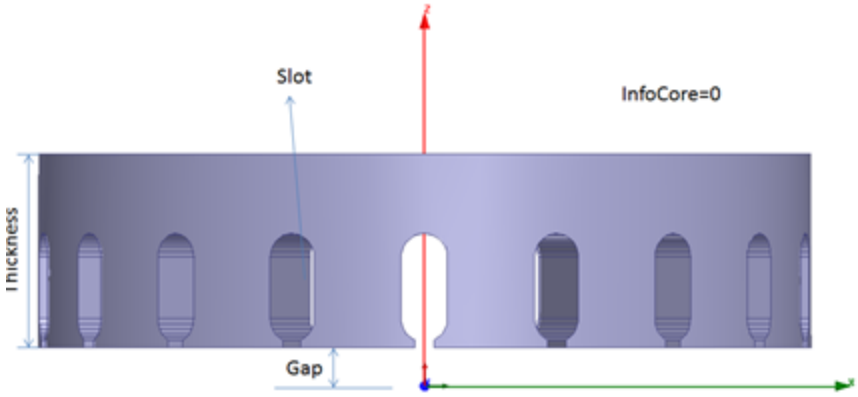


Figure 9-187 3D DiskSlot Core (InfoCore set to 0)

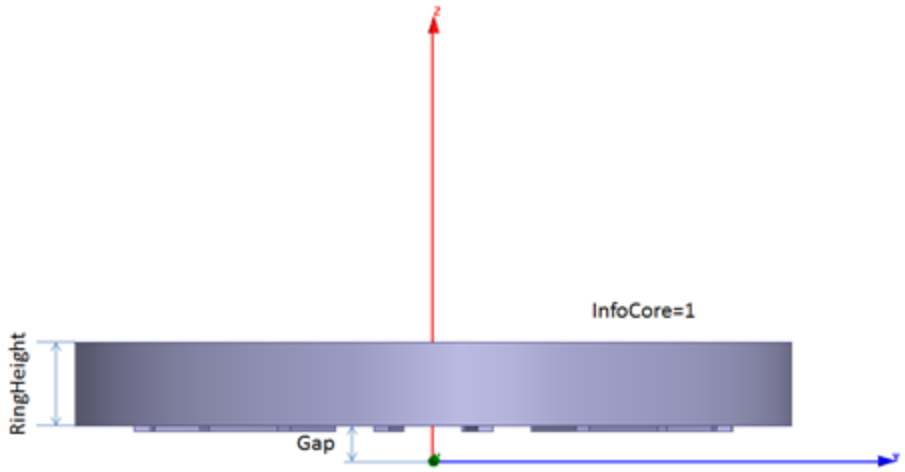


Figure 9-188 3D DiskSlot Core (InfoCore set to 1)

Creating an Upper or Lower Axial-FluxSlot Core (DiskSlot Core)

You can create an axial-flux slot core of a DiskSlot Core manually by setting the value of **InfoCore** to 0.

For an upper core, set the value of **Gap** to be larger than 0, as shown in Figure 9-184 and Figure 9-187 ; for a lower core, set this value to be smaller than 0, which is shown in Figure 9-189 .

You can assign the value of this property either when creating a DiskSlot core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

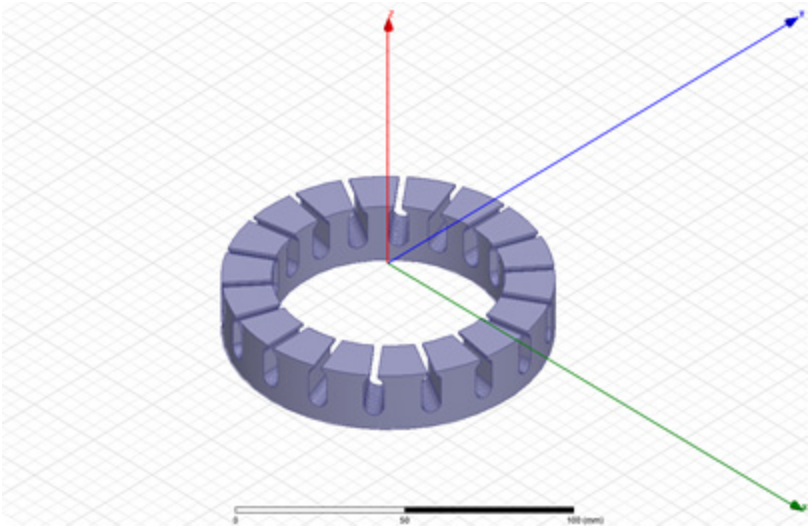


Figure 9-189 Lower axial-flux slot core

Creating a Squirrel-Cage Winding

You can create squirrel-cage winding of a DiskSlot core manually by setting the property of **InfoCore** to 1. An example of squirrel-cage winding is shown in Figure 9-186 .

You can assign the value of this property either when creating an axial-flux slot core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

DiskCoil UDP

The DiskCoil UDP is used to create a lap-type coil in an axial-flux slotted core created by the DiskSlotCore UDP for axial-flux machines. It can also create a coil terminal for current assignment.

Property	Description
DiaOuter	Core outer diameter.
DiaInner	Core inner diameter.
Thickness	Core axial thickness per side.
Gap	Gap between core & xy plane (>0: upper; <0: lower; =0: double-sided).
Skew	Skew angle.
Slots	Number of slots.
SlotType	Slot type: 1 to 7.
Hs0	Slot opening height.
Hs1	Slot wedge height.
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.
FilletType	0: a quarter circle; 1: tangent connection.
Layers	Number of winding layers.
CoilPitch	Coil pitch measured in slots.
EndExt	One-side radial end extended length.
EndClr	Clearance between two end spans; 0 for no span.
SegAngle	Deviation angle for end span (5~15, <5 for true surface).
InfoCoil	0: winding; 1: coil; 2: terminal.

Note	• For parameters such as **Skew**, **FilletType**, **SegAngle**, **HalfSlot** and parameters

	related to slots, refer to SlotCore UDP parameters. • Figure 9-191 can be used in comparison to Figure 9-190 to show **CoilPitch** parameter. • A DiskSlot core is used as reference in Figure 9-192 to show **EndExt** and **EndClr** parameters.

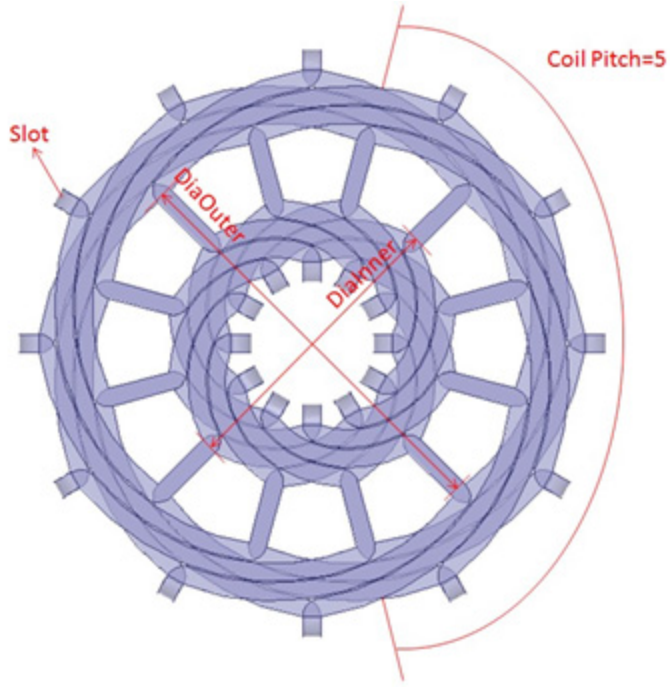


Figure 9-190 All disk coils (InfoCore set to 0)

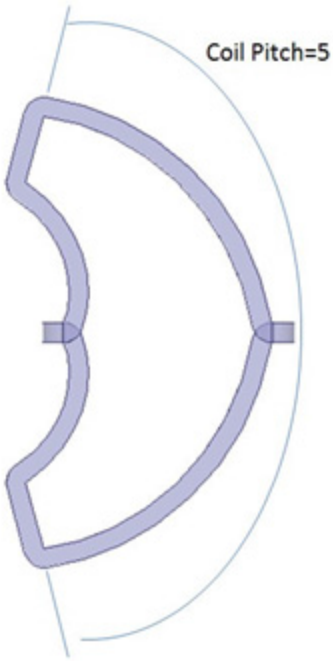


Figure 9-191 One disk coil (InfoCore set to 1)

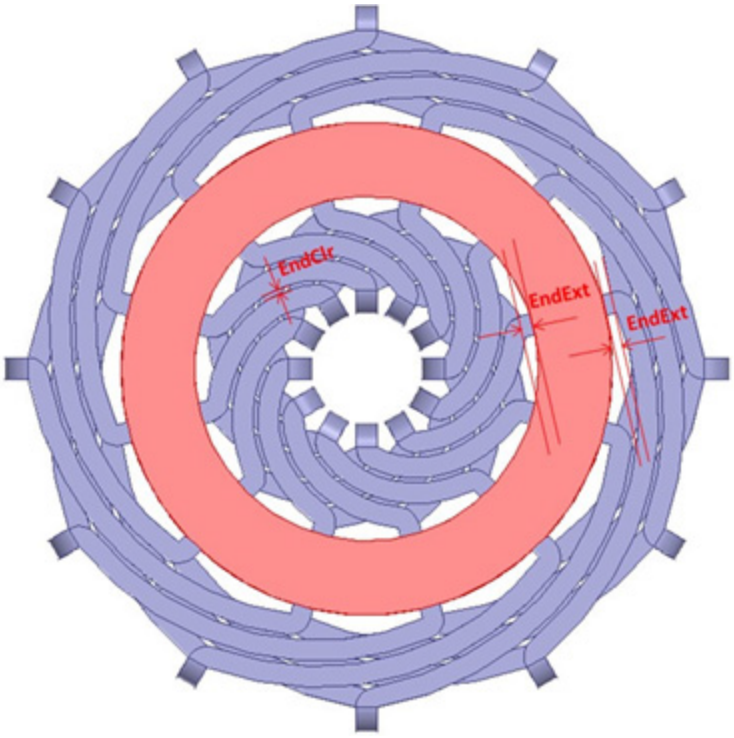


Figure 9-192 EndExt and EndClr parameters (InfoCore set to 0)

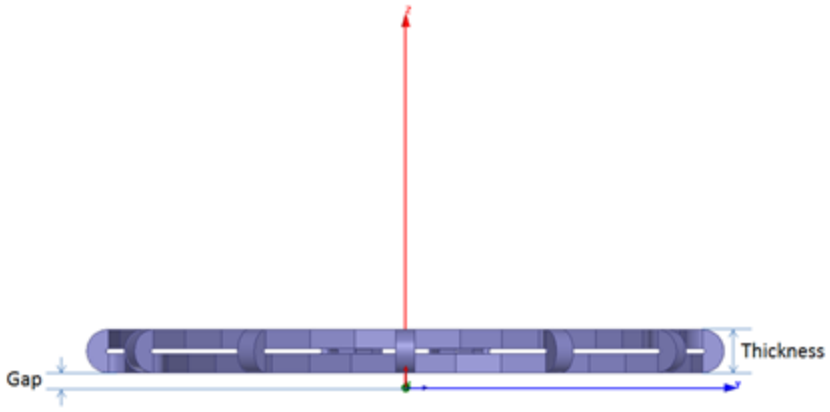


Figure 9-193 Gap and Thickness parameters (InfoCore set to 0)

Creating a Whole Winding (Upper or Lower)

You can create a whole winding of a DiskCoil Core manually by setting the property of **InfoCoil** to 0.

For an upper core, set the value of **Gap** to be larger than 0, which is shown in Figure 9-193 ; for a lower core, set this value to be smaller than 0, which is shown in Figure 9-194 .You can assign the value of this property either when creating the DiskCoil or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

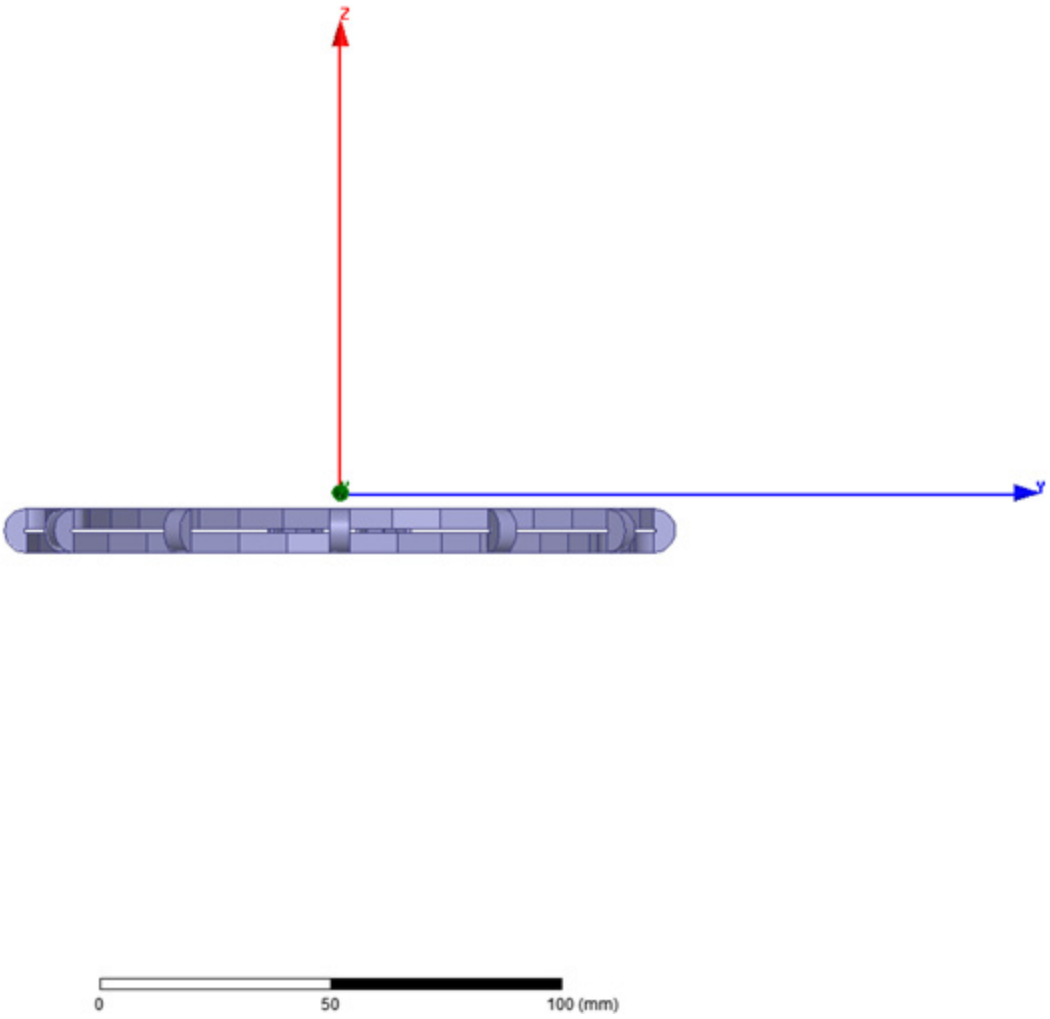


Figure 9-194 Lower whole winding

Creating a Lap Coil and its Terminals

You can create a lap coil of a DiskCoil manually by setting the value of **InfoCoil** to 1. An example is shown in Figure 9-191 .

You can create a terminal of a DiskCoil manually by setting the value of **InfoCoil** to 2. An example is shown in Figure 9-195 .

You can assign the value of this property either when creating a DiskCoil or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

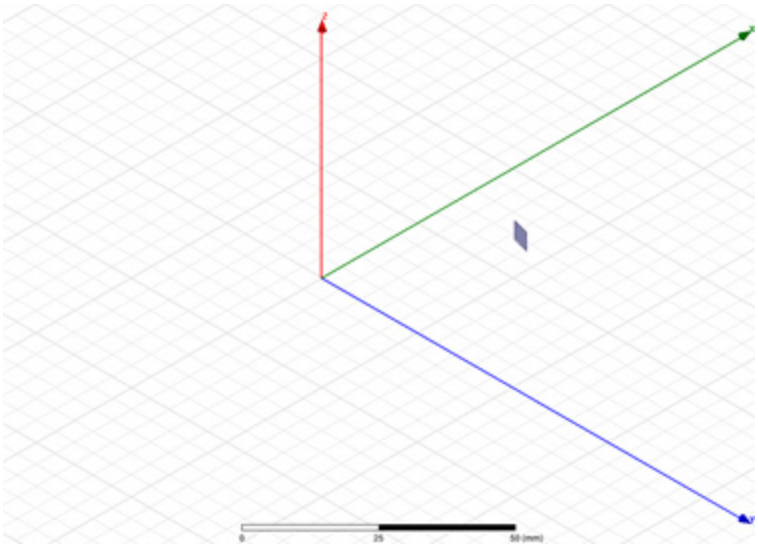


Figure 9-195 A terminal of a DiskCoil UDP

DiskPMCore UDP

The DiskPMCore UDP is used to create a PM core of the axial-flux type for axial-flux PM machines.

Property	Description
DiaOuter	Core outer diameter.
DiaInner	Core inner diameter.
Thickness	Core axial thickness per side.
Gap	Gap between core & xy plane (>0: upper; <0: lower; =0: double-sided).
Skew	Skew angle.
Poles	Number of poles.
Embrace	Pole embrace.
ThickMag	Magnet axial thickness per side.
InfoCore	0: core only; 1: all PMs; 2: one PM only.

Note	There are two objects in each of the following figures. Both objects in each figure are DiskPMCore. The value of **InfoCore** of one DiskPMCore is 1 (red object), the other's is 0 (gray object).

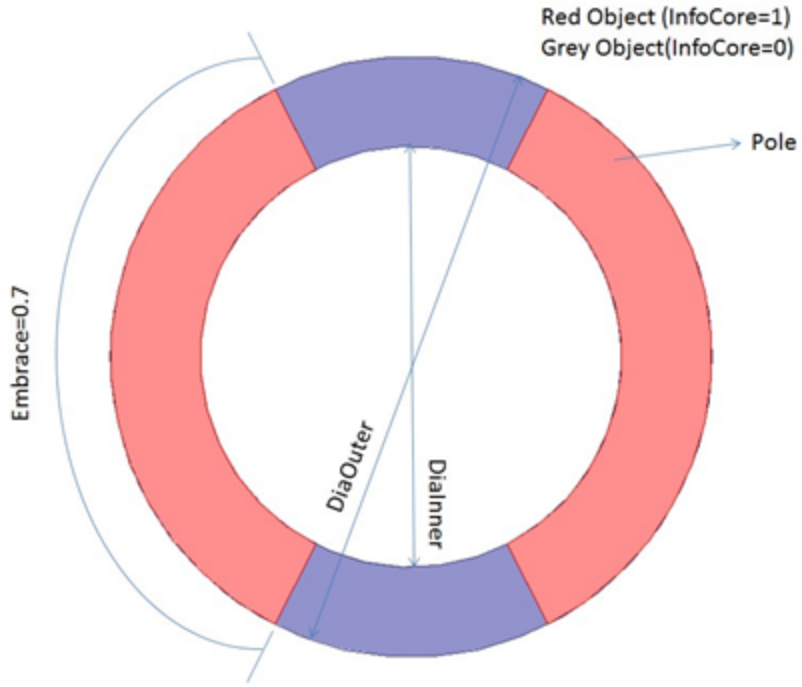


Figure 9-196 DiskPM core parameters on x-y plane

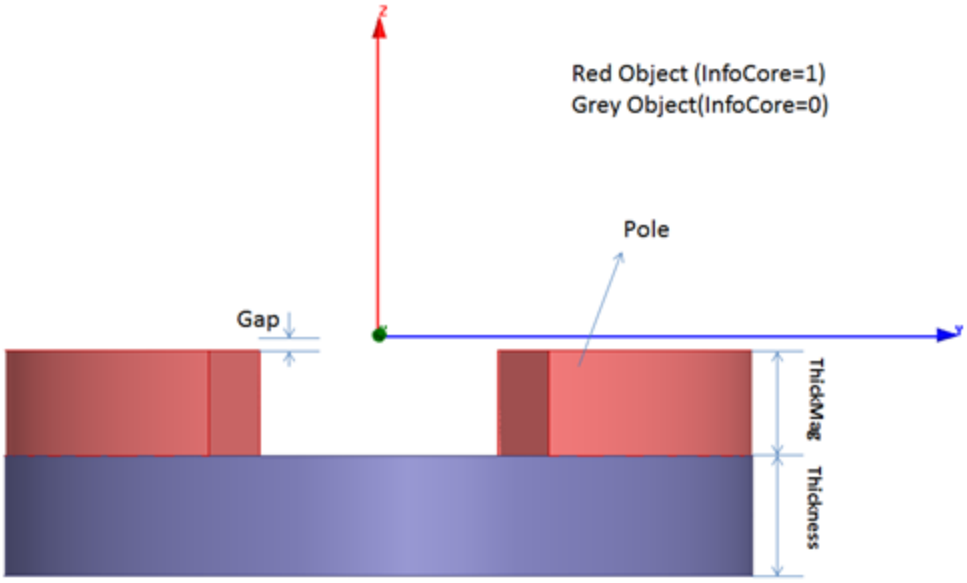


Figure 9-197 DiskPM core parameters in axial direction

Creating an Upper or Lower Axial-Flux PM Core

You can create an axial-flux PM core of a DiskPM core manually by setting the value of **InfoCore** to 0.

For an upper core, set the value of **Gap** to be larger than 0; for a lower core, set this value to be smaller than 0.

An example of an upper core is shown in Figure 9-198 , whereas a lower core is in Figure 9-199 .

You can assign the value of this property either when creating a Disk PM core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

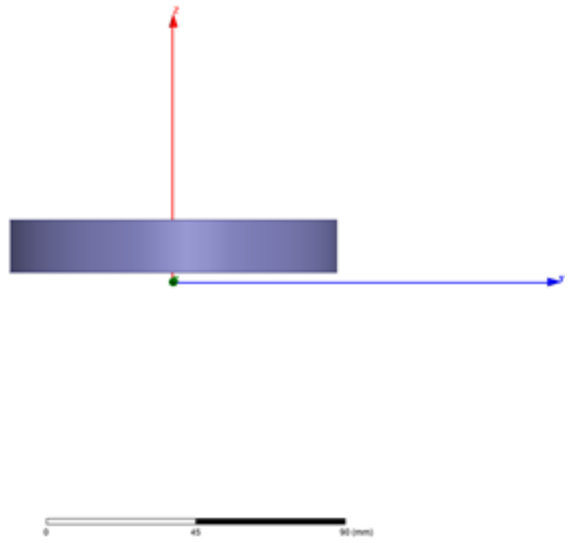


Figure 9-198 . Upper axial-flux PM core

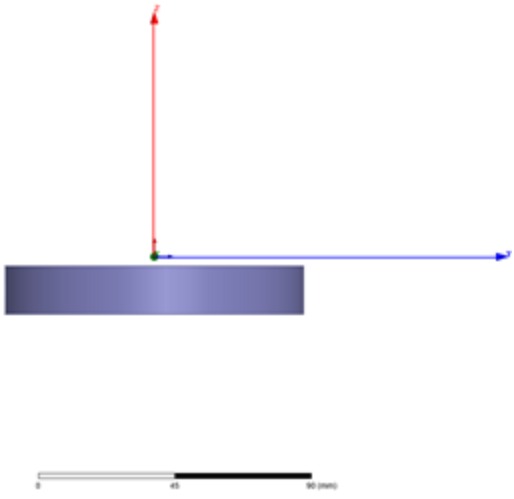


Figure 9-199 Lower axial-flux PM core

Creating All Magnets (Disk PM Core)

You can create all magnets of a disk PM core manually by setting the value of **InfoCoil** to 1. An example of all magnets is shown in Figure 9-196 and Figure 9-197 as a red object.

You can assign the value of this property either when creating a disk PM core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Creating One Magnet Only (Disk PM Core)

You can create a single magnet of a disk PM core manually by setting the property of **InfoCoil** to 2. An example of one magnet is shown in Figure 9-200 .

You can assign the value of this property either when creating a disk PM core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

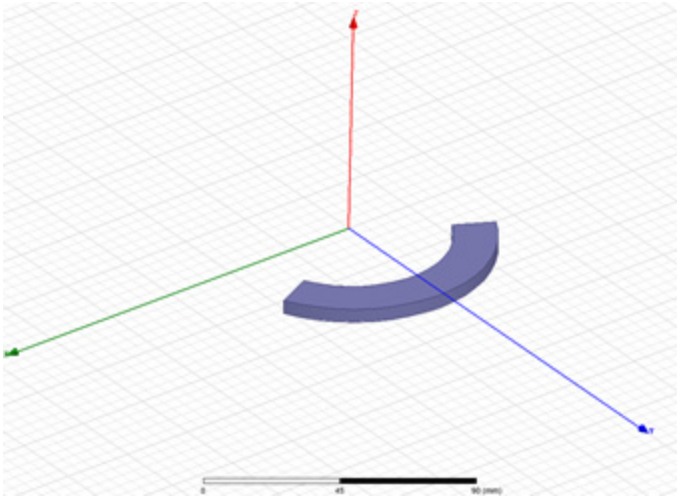


Figure 9-200 One magnet of disk PM core

LinearMCore UDP

The LinearMCore UDP is used to create the mover core and the armature winding for linear motors. It can also create a coil terminal for current assignment.

Property	Description
WidthCore	Core width in motion direction.
ThickCore	Core thickness.
Length	Core length.
SlotPitch	The distance between two slots.

Property	Description
Slots	Number of slots.
SlotType	Slot type: 1 to 7.
Hs0	Slot opening height.
Hs1	Slot wedge height.
Hs2	Slot body height.
Bs0	Slot opening width.
Bs1	Slot wedge maximum width.
Bs2	Slot body bottom width, 0 for parallel teeth.
Rs	Slot body bottom fillet.
Layers	Number of winding layers.
CoilPitch	Coil pitch measured in slots.
EndExt	One-side end extended length.
SpanExt	Axial length of end span; 0 for no span.
SegAngle	Deviation angle for slot arches (10~30, <10 for true surface).
InfoCore	0: core only; 1: core & all coils; 2: one coil only.

Note	
	• For parameters such as **SegAngle** and parameters related to slots, refer to SlotCore UDP parameters. • The InfoCore values of the following two figures are both 1 to better describe other parameters such as **EndExt** and **SpanExt**.

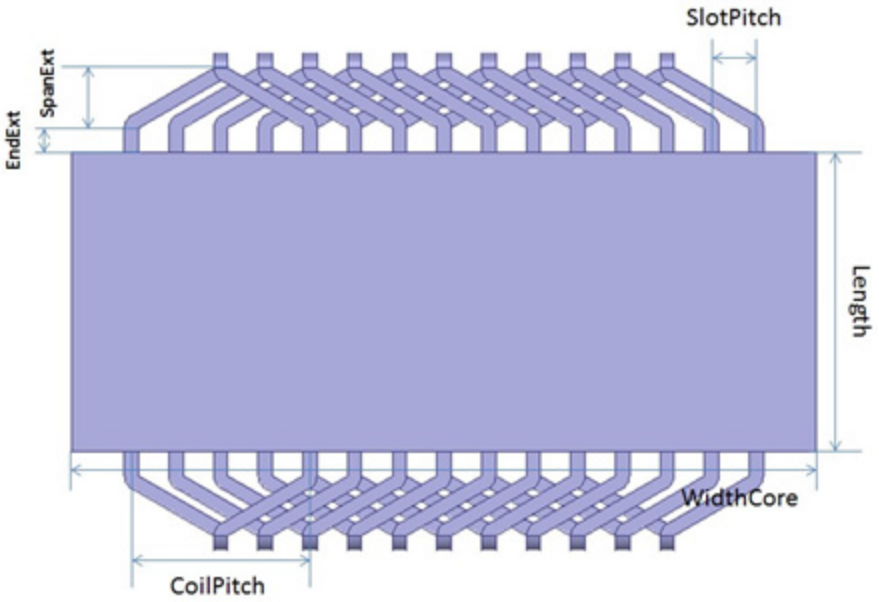


Figure 9-201 Parameters of LinearMCore in x-y plane (InfoCore set to 1)

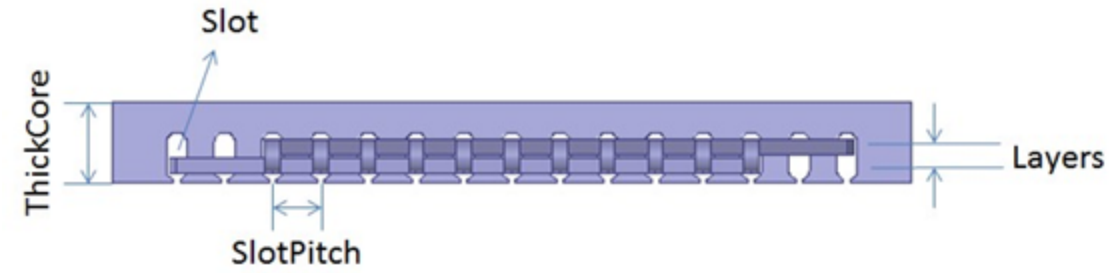


Figure 9-202 Parameters of LinearMCore in axial direction (InfoCore set to 1)

Creating a Linear Machine Core

You can create a linear machine core manually by setting the value of **InfoCore** to 0.

An example of a linear machine core is shown in Figure 9-203 .

You can assign the value of this property either when creating a linear-machine core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

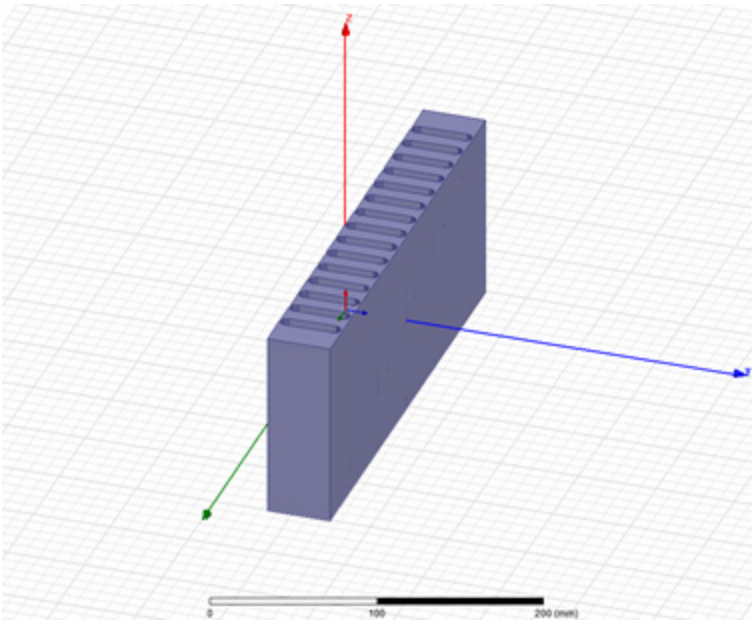


Figure 9-203 Linear Machine Core (InfoCore set to 0)

Creating a Core and All Coils (Linear Machine Core)

You can create a core and all coils manually by setting the value of **InfoCore** to 1. An example of a core and all coils of a linear-machine core is shown in Figure 9-204 .

You can assign the value of this property either when creating a linear machine core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

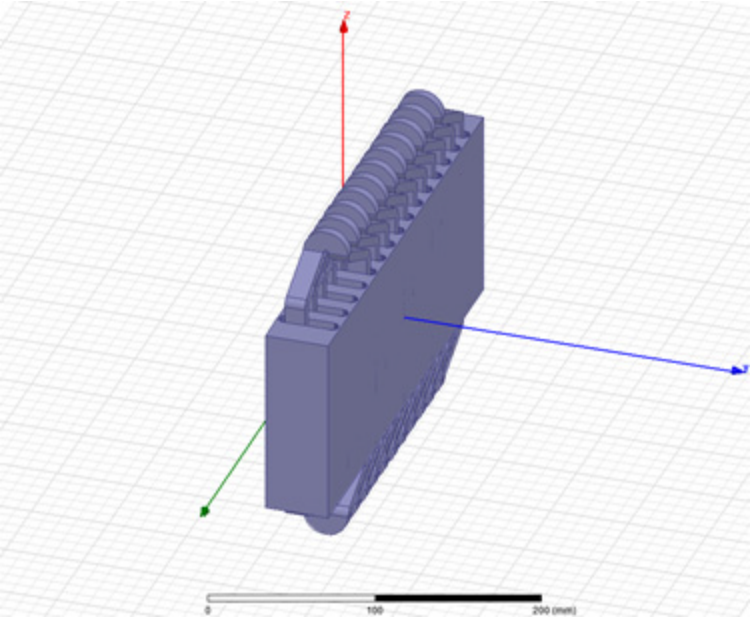


Figure 9-204 A core and all coils of a linear machine core (InfoCore set to 1)

Creating One Coil Only (Linear Machine Core)

You can create a one coil only manually by setting the value of **InfoCore** to 2.

An example of a coil of a linear machine core is shown in Figure 9-205 .

You can assign the value of this property either when creating a linear machine core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

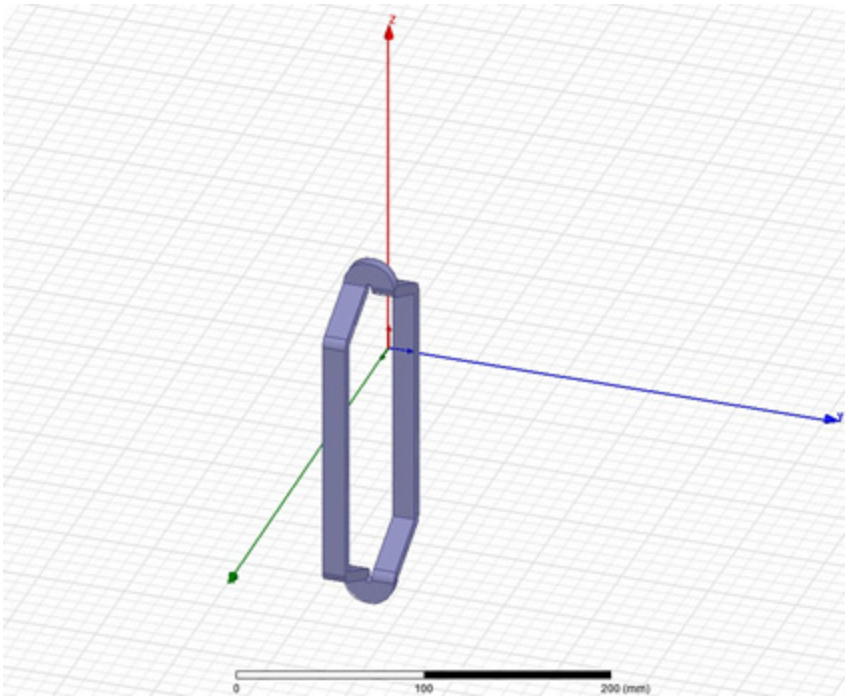


Figure 9-205 One coil of a linear machine core (InfoCore set to 2)

TransCore UDP

The TransCore UDP is used to create a three-leg transformer core for three-phase transformers.

Property	Description
DiaLeg	Outer diameter of leg cross-section.
DistLeg	Leg center to center distance.
DistYoke	Yoke center to center distance.
Stages	Number of stages of leg cross-section.
ThickCore	Core thickness, only used for Stages = 1.
WidthYoke	Yoke width, =0 for same cross-section as leg's.
InfoCore	0: whole core; 1: legs only; 2: yokes only.

Note	Figure 9-206 , Figure 9-207 , and Figure 9-208 show a TransCore UDP with different values of **Stages** and **WidthYoke**.

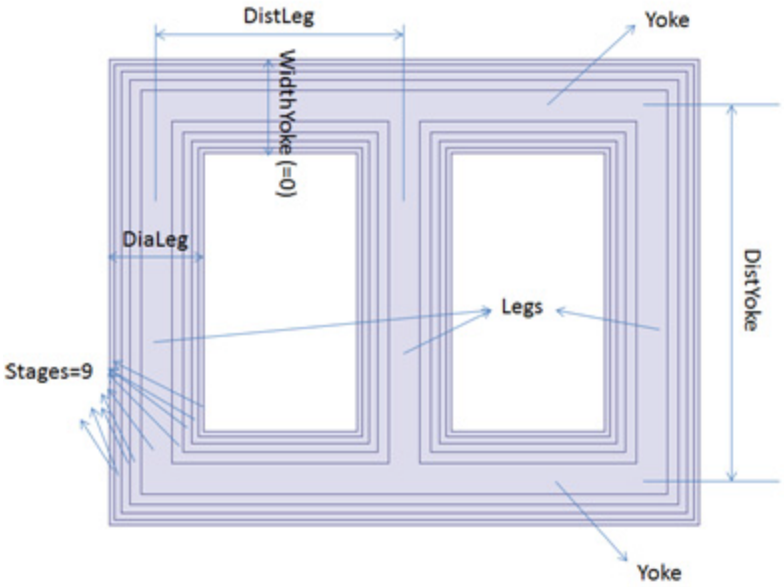


Figure 9-206 TransCore when Stages is not 1 and WidthYoke is 0

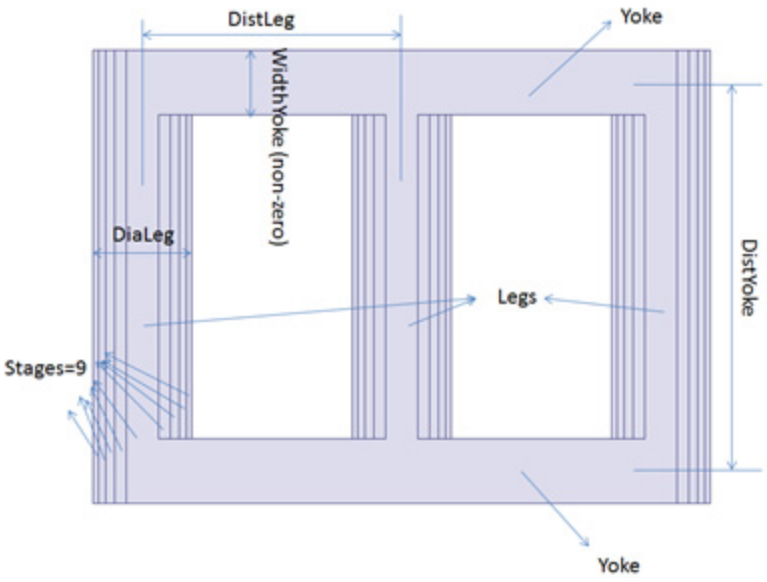


Figure 9-207 TransCore when Stages is not 1 and WidthYoke is not 0

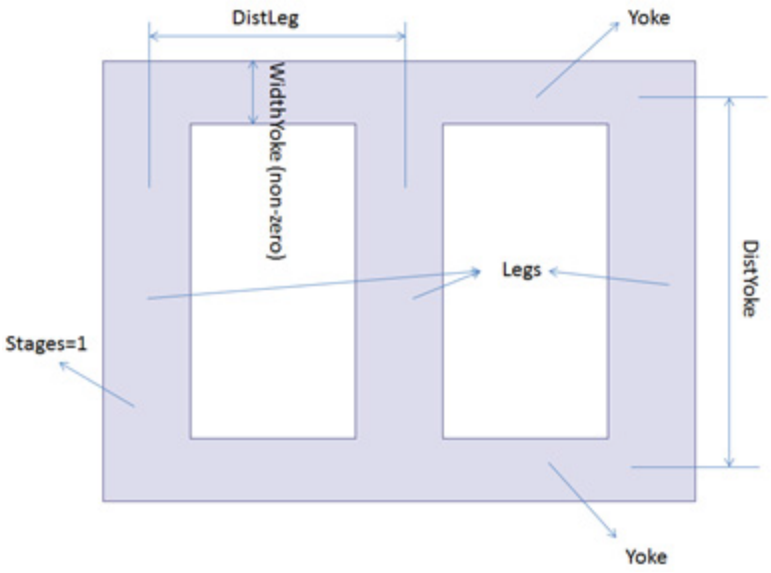


Figure 9-208 TransCore when Stages is 1 and WidthYoke is not 0

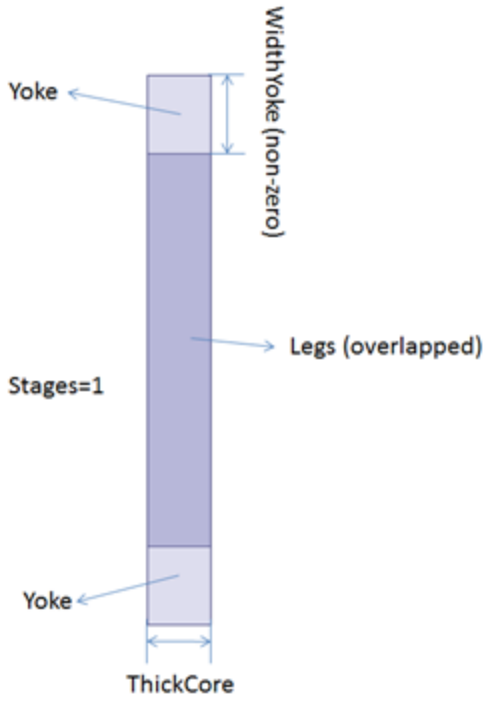


Figure 9-209 Parameters of TransCore in axial direction (when Stages is 1 and WidthYoke is not 0)

Creating a Three-Leg Transformer Core

You can create a three-leg transformer core manually by setting the value of **InfoCore** to 0.

Examples of a three-leg transformer core are shown in Figure 9-206 , Figure 9-207 , Figure 9-208 , and Figure 9-209 .

You can assign the value of this property either when creating a three-leg transformer or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Creating Three Legs Only

You can create three legs of the core manually by setting the value of **InfoCore** to 1.

An example of three legs is shown in Figure 9-210 .

You can assign the value of this property either when creating a three-leg transformer core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

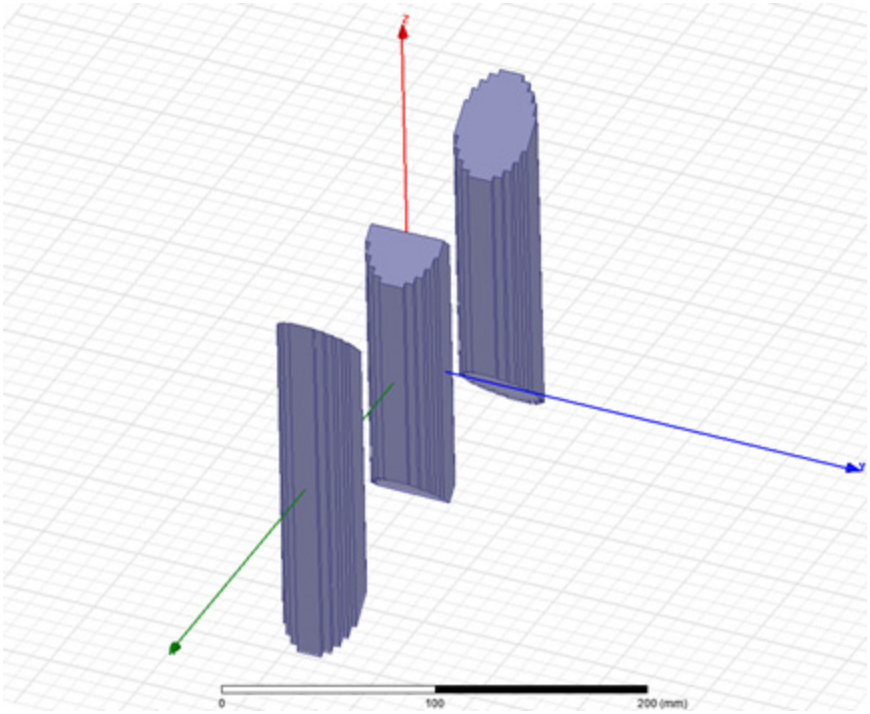


Figure 9-210 Three legs of TransCore (InfoCore set to 1)

Creating Two Yokes Only

You can create two yokes of the core manually by setting the value of **InfoCore** to 2.

An example of two yokes is shown in Figure 9-211 .

You can assign the value of this property either when creating a three-leg transformer core or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

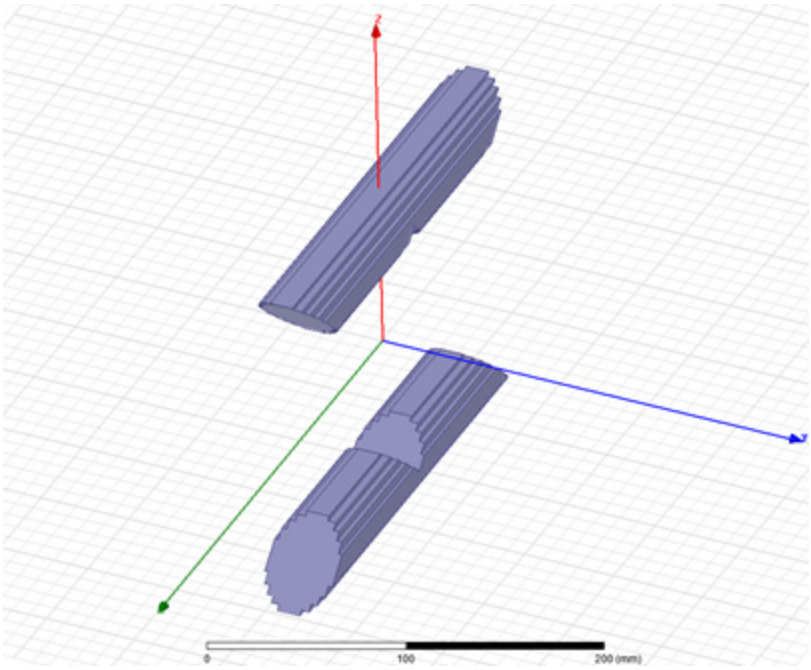


Figure 9-211 Two Yokes of TransCore (InfoCore set to 2)

TransCoil UDP

The TransCoil UDP is used to create a coil in a three-leg transformer core created by the TransCore UDP for three-phase transformers. It can also create a coil terminal for current assignment.

Property	Description
DistLeg	Leg center to center distance.
CoilType	Coil type: 1 for solenoid coil; 2 for pancake coil.
WidthIn	Coil width between two inner sides.
DepthIn	Coil depth between two inner ends.
RadiusIn	Coil inner fillet radius.
ThickCoil	Coil thickness of one side.
HighCoil	Coil height.
Layers	Number of layers.
GapLayer	Gap between two layers.
InfoCore	0: all coils; 1: one coil only.

Note	The key to making the cross-section of each layer of a coil to be a circle is to set the value of **WidthIn** equal to that of **DepthIn** and at the same time set the value of **RadiusIn** to be half of that of **WidIn**. That is, **WidthIn**=**DepthIn**=2***RadiusIn**. The cross-section of a layer could be a non-circle if the value of **WidthIn** is not equal to that of **DepthIn** or **RadiusIn**'s value is not half of **WidthIn**'s. Figure 9-214 shows a situation where a layer is a non-circle.

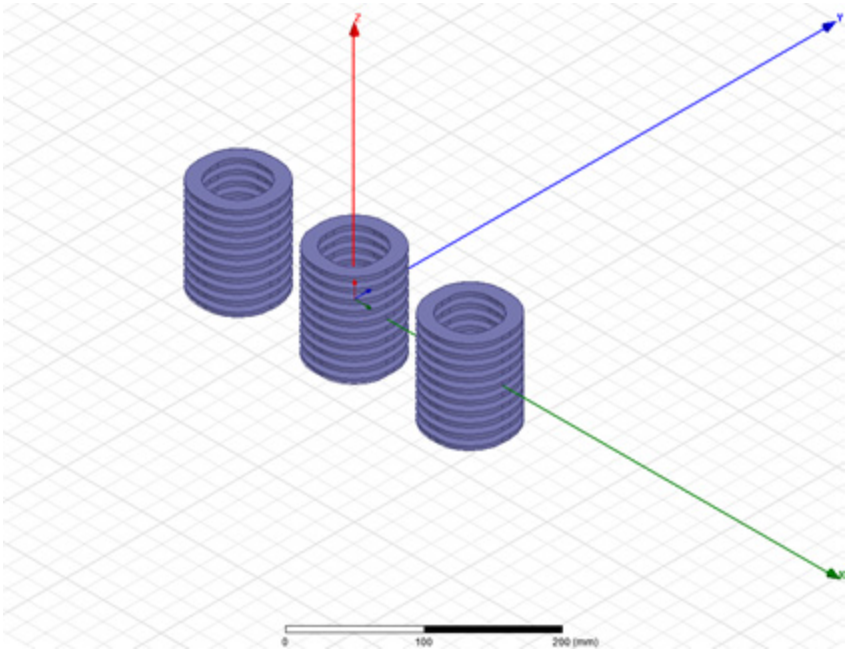


Figure 9-212 Overview of TransCoil (InfoCore set to 0)

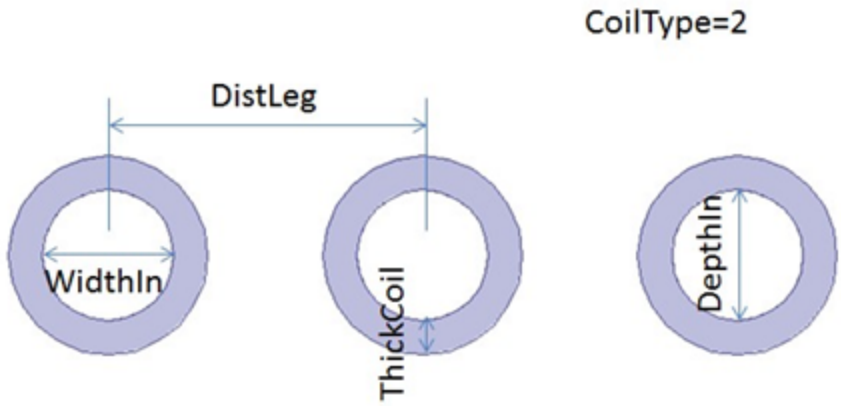


Figure 9-213 Parameters of TransCoil in x-y plane (InfoCore set to 0)

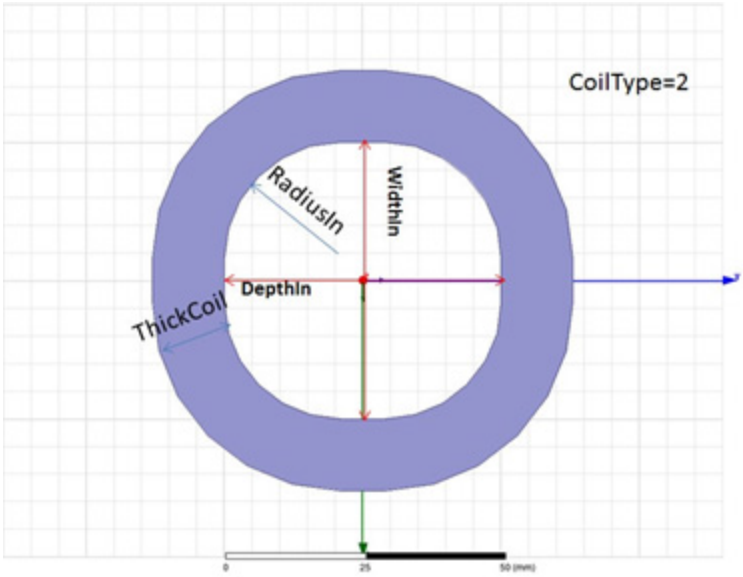


Figure 9-214 RadiusIn of TransCoil

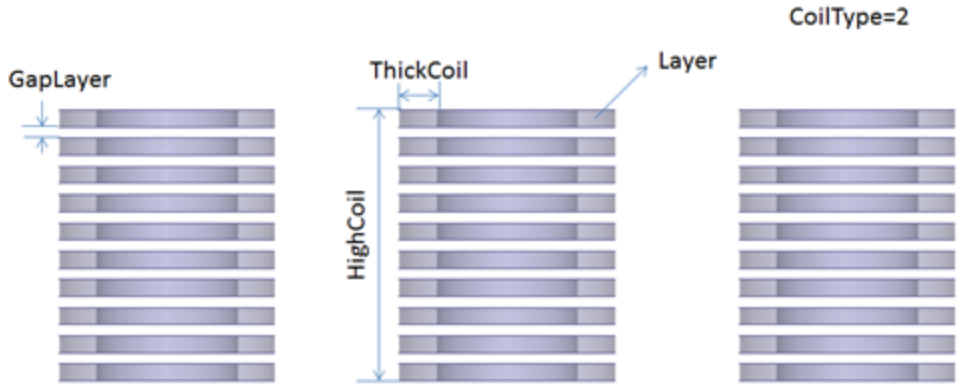


Figure 9-215 Parameters of TransCoil in axial direction (InfoCore set to 0)

Creating Coils for All Three Legs

You can create coils for three legs manually by setting the value of **InfoCore** to 0.

Examples of coils for TransCoil UDP are shown in Figure 9-212 , Figure 9-213 , and Figure 9-215 .

You can assign the value of this property either when creating a TransCoil UDP or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

Creating One-Leg Coil Only

You can create a one-leg coil manually by setting the value of **InfoCore** to 1.

An example of a one-leg coil for a TransCoil UDP is shown in Figure 9-216 .

You can assign the value of this property either when creating a TransCoil UDP or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

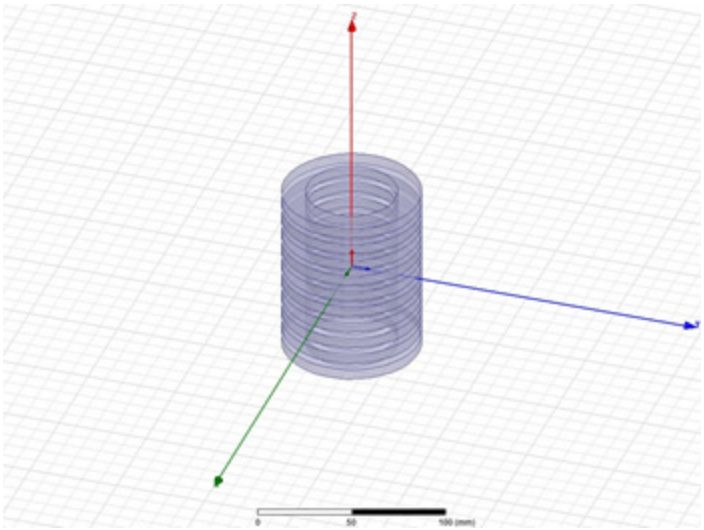


Figure 9-216 One-leg coil for TransCoil UDP

TransTwist UDP

Property	Description
CoreType	1: shell-type; 2: core-type.
WidthOut	Core outer width.
DepthOut	Core outer depth in coil turn direction.
RadiusOut	Core outer fillet radius.
WidthIn	Core window width.
DepthIn	Core window depth in coil turn direction.
RadiusIn	Core window fillet radius.
DistCenters	Window center-to-center distance, for shell-type only.
HeightCore	Core height.
Coils	Number of coils: 1 or 2.
WiresCoil	Wires per coil: 1, 2, 3, or 4 (no twist for 1 wire).

Property	Description
TurnsCoil	Number of turns per coil.
TurnsTwist	Total twist turns in one coil turn; 360 degrees in one twist turn.
DiaWire	Wire diameter.
Segments	Polygon segments for wire cross section: <=2 for true circle.
InsType	Insulator type: 1: round insulator; 2: filleted square insulator.
ThickIns	Outer minimum thickness of insulator.
GapWires	The gap between two round wire profiles.
GapEndTurns	The gap between two end turns of round insulator profiles.
GapCoreIns	The gap between core and round insulator profiles inside two windows and at two ends.
LeadExt	Lead extension from lead terminal to core end terminal.
InfoCore	0: All; 1: Core; 2: Insulator; 3-6: Wires of coil#1; 7-10: Wires of coil#2.
InfoTerm	0: 3D wire; 1: 2D go-term; 2: 2D re-term (for InfoCore = 3-10 only).

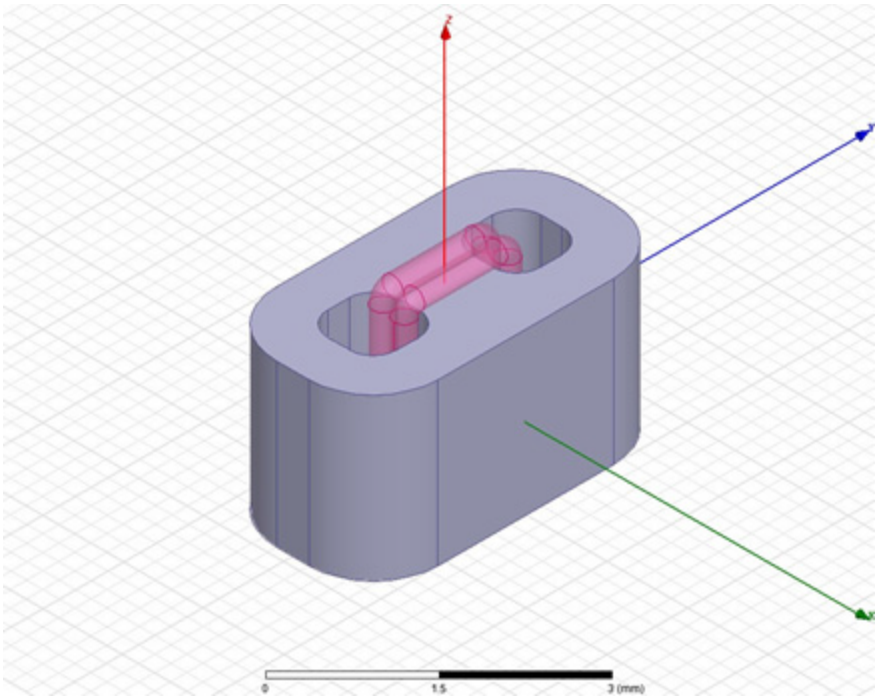


Figure 9-217 Overview of TransTwist UDP (without wires)

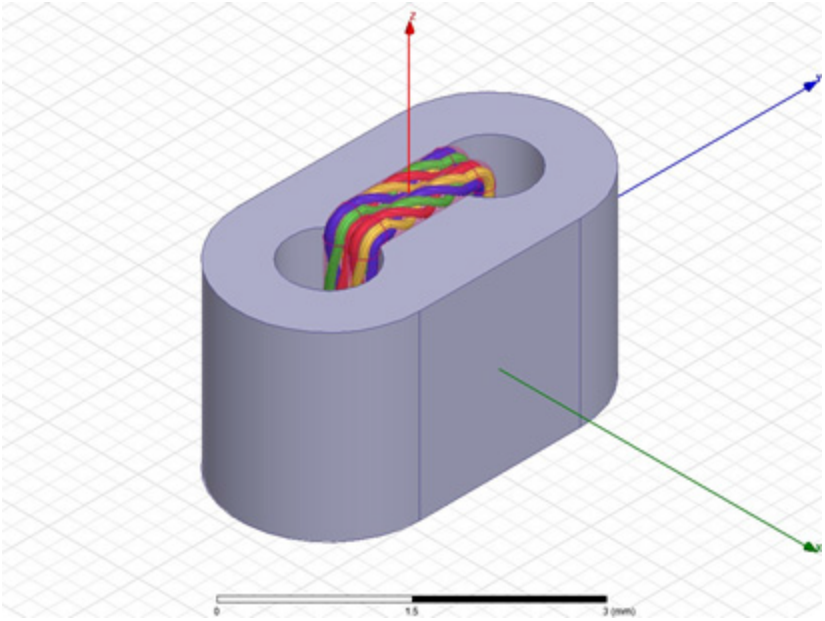


Figure 9-218 Overview of TransTwist UDP (with colorful wires)

Note	• Figure 9-217 and Figure 9-218 give an overview of the TransTwist UDP. The difference between them is Figure 9-218 has 4 colorful wires inside the insulator whereas Figure 9-217 has nothing in its insulator. • InFigure 9-217 and Figure 9-218 and also other figures like Figure 9-222 , objects in different colors are all TransTwist UDP but with different **InfoCore** values. For example, in Figure 1-6 , the gray object is a TransTwist UDP whose **InfoCore** value is 1(core), while red, blue, green and yellow objects (wires) are also TransTwist UDPs whose **InfoCore** values are 3, 4, 5, and 6 respectively. Also, the value of the **InfoCore** of pink tube that covers the wires is 2 (insulator). To draw this kind of figure, copy and paste the TransTwist UDP (or manually create multiples of this UDP) and then change their **InfoCore** values and colors. • The reason for making colorful figures here is to show the parameters of the TransTwist UDP more clearly. • The difference between Figure 9-220 and Figure 9-221 is the parameter **TurnsCoil**, which means the number of turns for each coil (only one coil in each figure). There are 2 turns for the coil in Figure 1-4 whereas there is only 1 turn for the coil in Figure 9-221 . • The purpose ofFigure 9-223 is to show the concept of the parameter **Coils**. Note that when the value of **Coils** is 2 in this figure, the two coils are disconnected. For a comparison, see the one coil shown in Figure 9-222 . • The purpose of Figure 9-224 is to show the parameter of **GapEndTurns**, which is useful only when the value of **TurnsCoil** is larger than 2. • The results of parameter **TurnsTwist** are difficult to show in a figure. It is used to define the number of twist turns in one coil turn for each coil (or wire). As shown in

	Figure 9-226 , the green wire is twisted 0.5 turn at the bottom of the core. Then, for the entire coil turn, it may be twisted 3 to 3.5 twist turns (0.5 to 1 turn on the top of the core, and 1 turn on each side of the core).

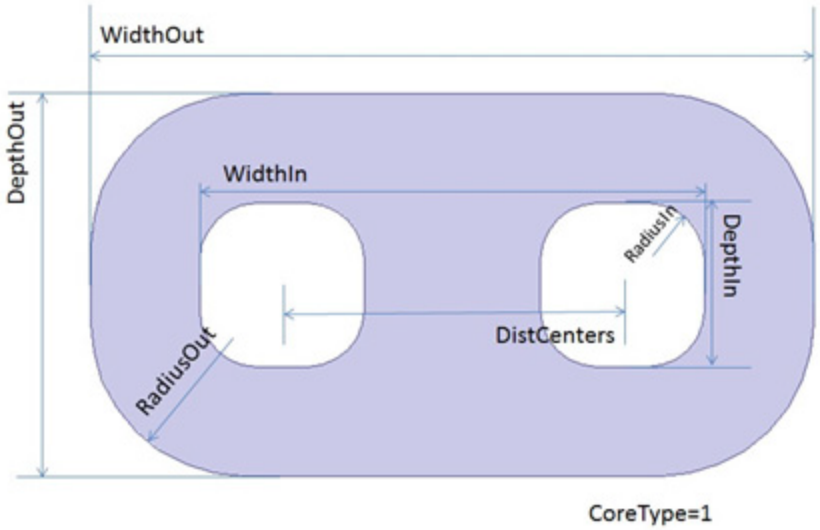


Figure 9-219 2D parameters of TransTwist UDP (InfoCore set to 1)

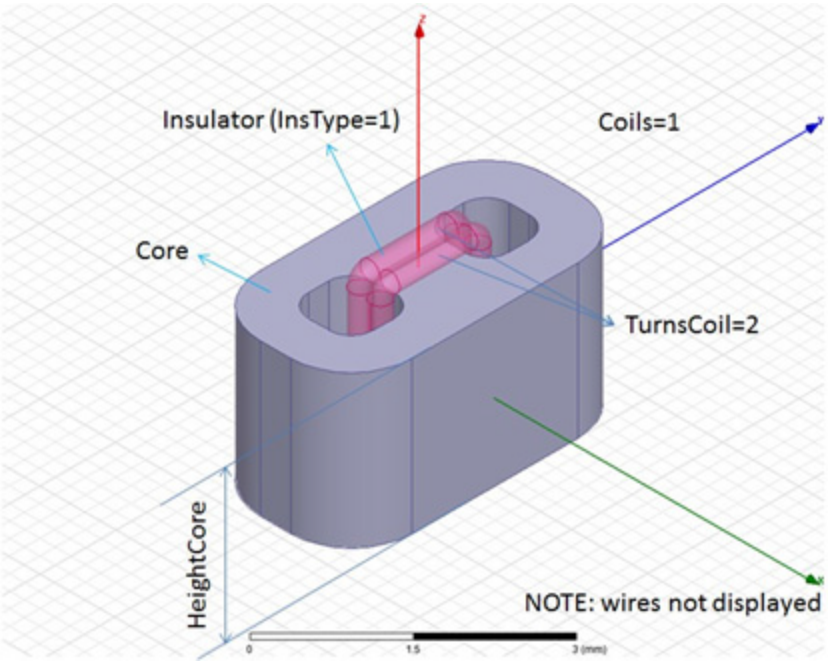


Figure 9-220 3D parameters of TransTwist UDP (gray Object: InfoCore set to 1, Red Object: InfoCore set to 2)

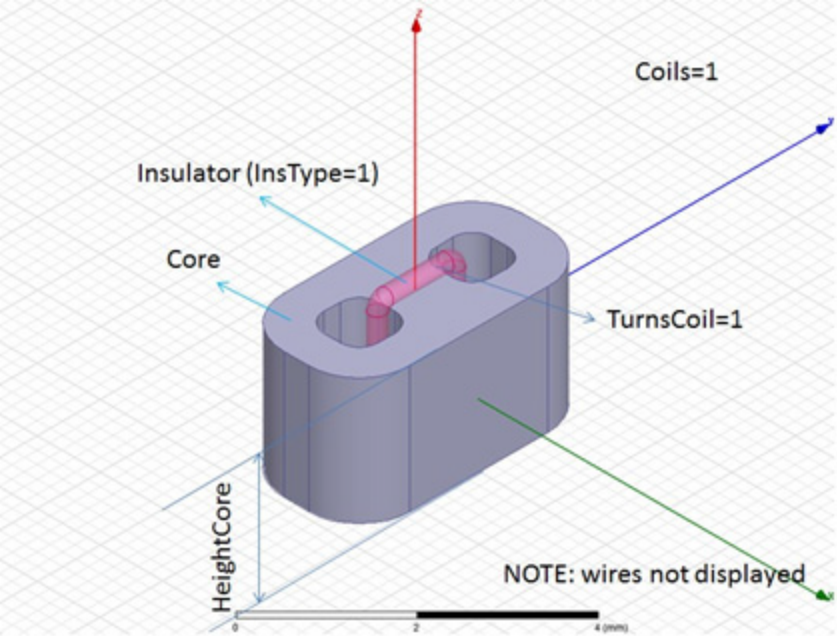


Figure 9-221 3D parameters of TransTwist UDP (gray Object: InfoCore set to 1, Red Object: InfoCore set to 2)

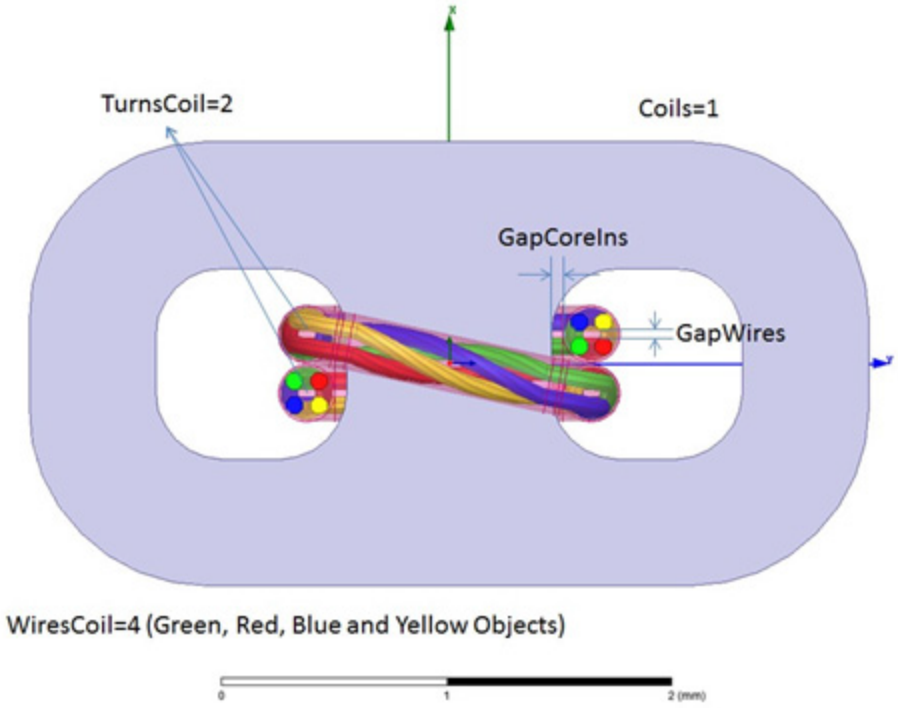


Figure 9-222 Gaps-related parameters of TransTwist UDP (gray Object: InfoCore set to 1, Red Object: InfoCore set to 2, Colorful Wires: InfoCore set to 3 to 6)

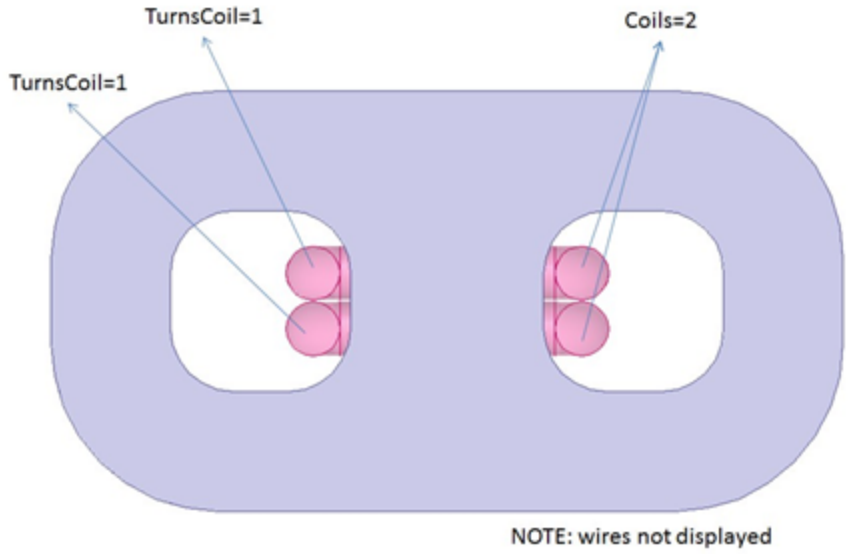


Figure 9-223 Coil-related parameters of TransTwist UDP (gray Object: InfoCore set to 1, Red Object: InfoCore set to 2)

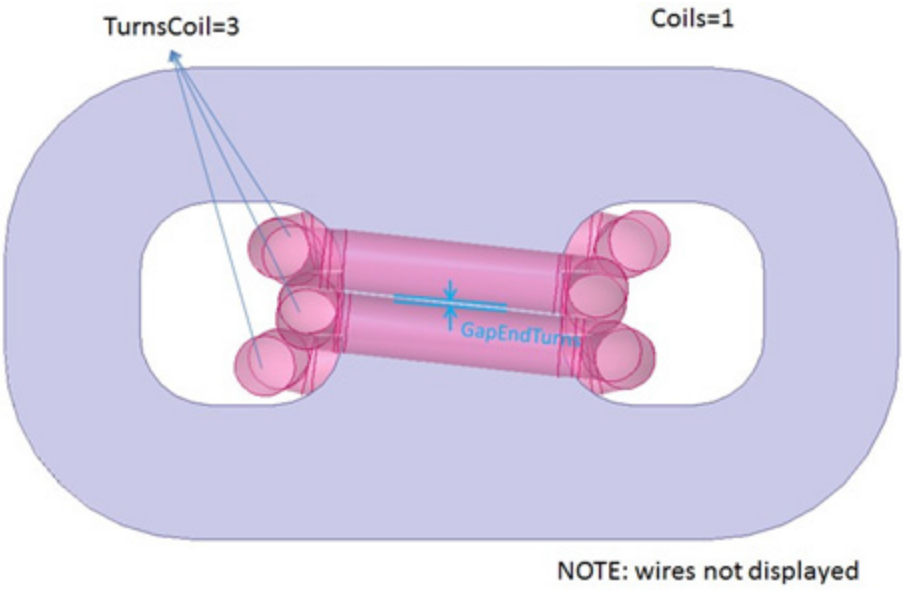


Figure 9-224 GapEndTurns of TransTwist UDP (gray Object: InfoCore set to 1, Red Object: InfoCore set to 2)

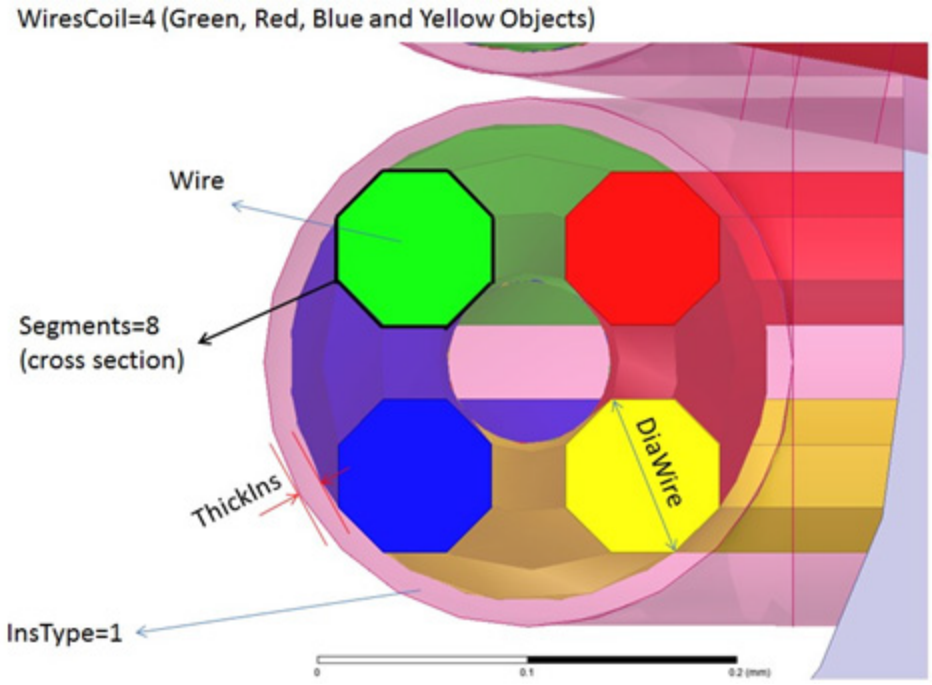


Figure 9-225 Wires-related parameters of TransTwist UDP (gray Object: InfoCore set to 1, Red Object: InfoCore set to 2, Colorful Wires: InfoCore set to 3 to 6)

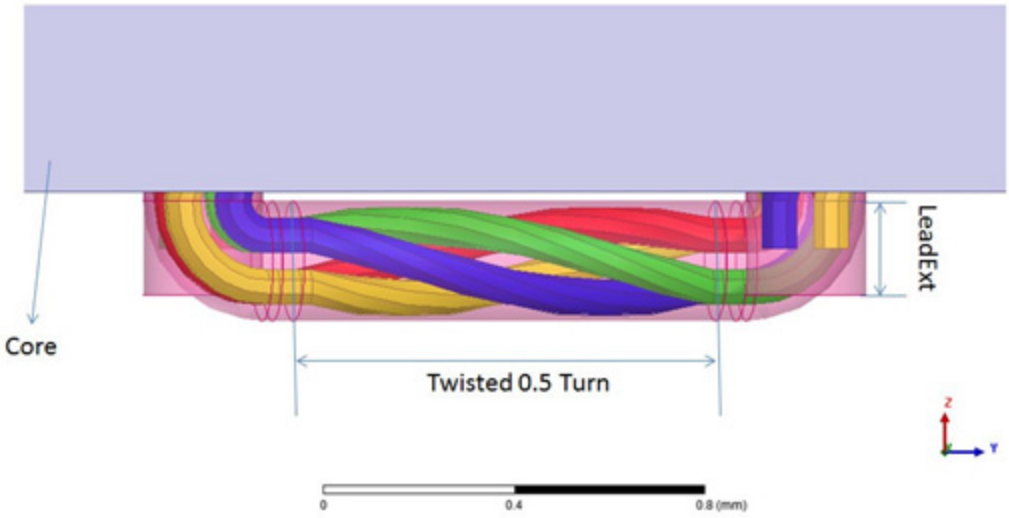


Figure 9-226 LeakEndExt of TransTwist UDP (gray Object: InfoCore set to 1, Red Object: InfoCore set to 2, Colorful Wires: InfoCore set to 3 to 6)

Creating a Core and All Twist Coils

You can create a core and all twist coils manually by setting the value of InfoCore to 0, which is set by default in Maxwell.

An example of a core and all twist coils is shown in Figure 9-227 .

You can assign the value of this property either when creating the UDP or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

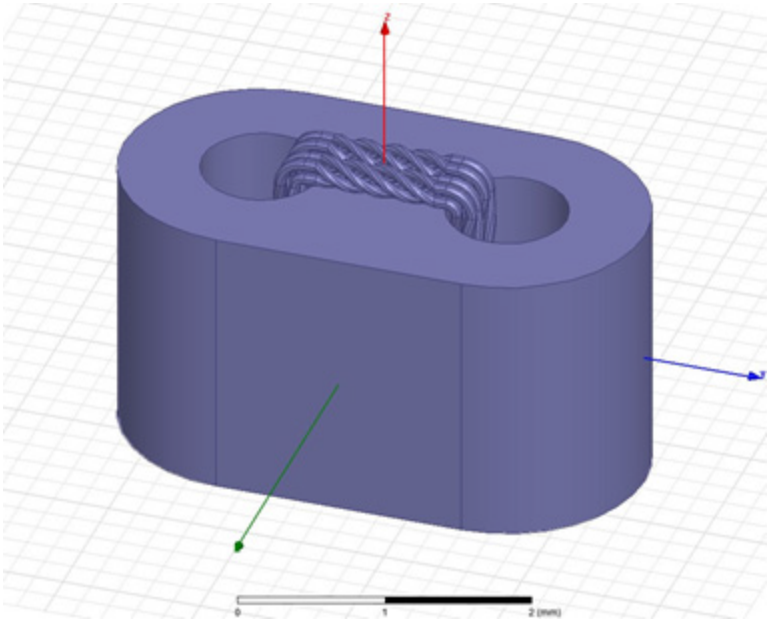


Figure 9-227 TransTwist UDP (InfoCore set to 0)

Creating a Twist Wire and its Terminals

You can create a twist wire of a TransTwist manually by setting the value of **InfoCore** to 3 to 6 (or 7-10 for coil #2) and **Info Term** to 0. Examples are shown in Figure 9-218 , Figure 9-221 , Figure 9-225 , and Figure 9-226 as red, green, blue and yellow objects.

You can create a terminal of a wire manually by setting the value of **InfoCore** to 3 to 6 (or 7-10 for coil #2) and **Info Term** to 1 or 2 (for 2D go-term and 2D re-term respectively). An example is shown in Figure 9-228 below.

You can assign the value of this property either when creating a TransTwist or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

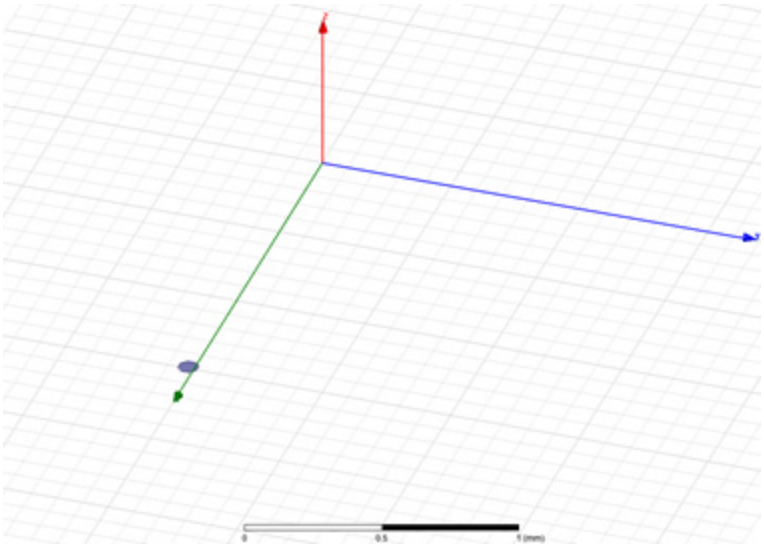


Figure 9-228 2D go-terminal of a wire

Creating a Twist Insulator of All Coils

You can create an insulator of all coils manually by setting the value of **InfoCore** to 2 and **Info Term** to 0. An example is shown in Figure 9-229 .

You can assign the value of this property either when creating a TransTwist or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

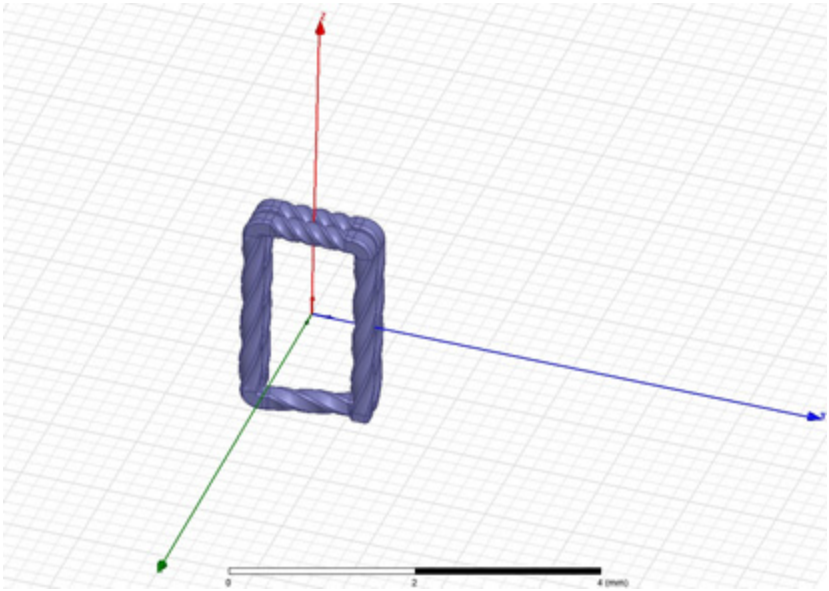
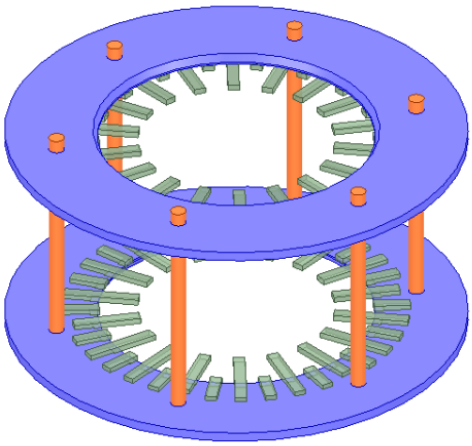


Figure 9-229 Twist insulator of all coils

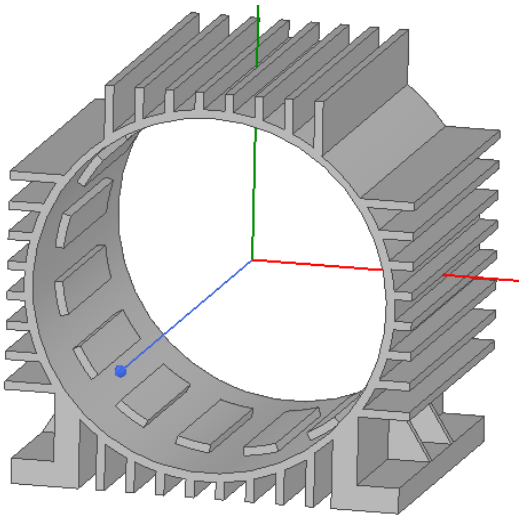
UDPs for Other Machine Parts

RMxprt includes UDPs to add various other types of machine parts. These include:

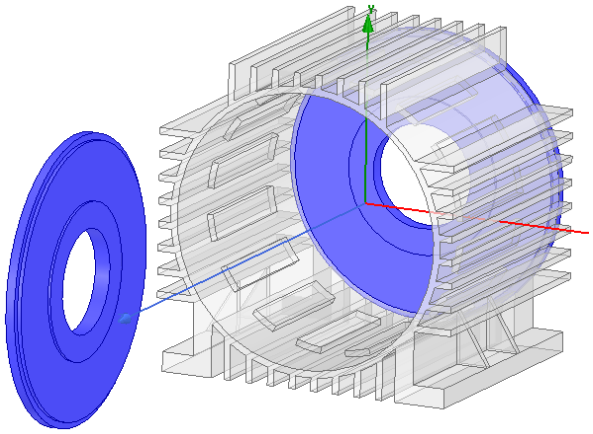
- PressPlate UDP - used to add press plates, fingers, and stems that are used to hold together the stack of core laminations in large AC and DC machines.



- Housing UDP - used to add machine housings and their associated fins, cooling ducts, and support feet.



- EndCap UDP - used to add the end caps and associated bearing surfaces for a housing.



Related Topics

Adding Machine Housings in RMxprt

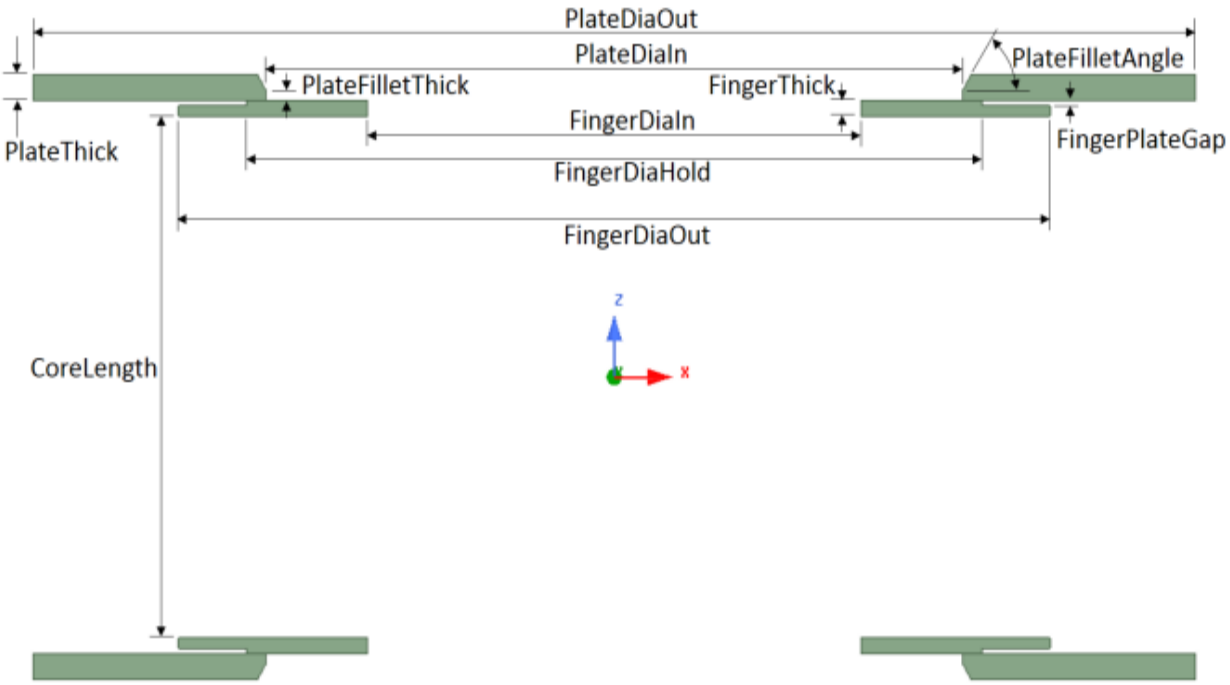
PressPlate UDP

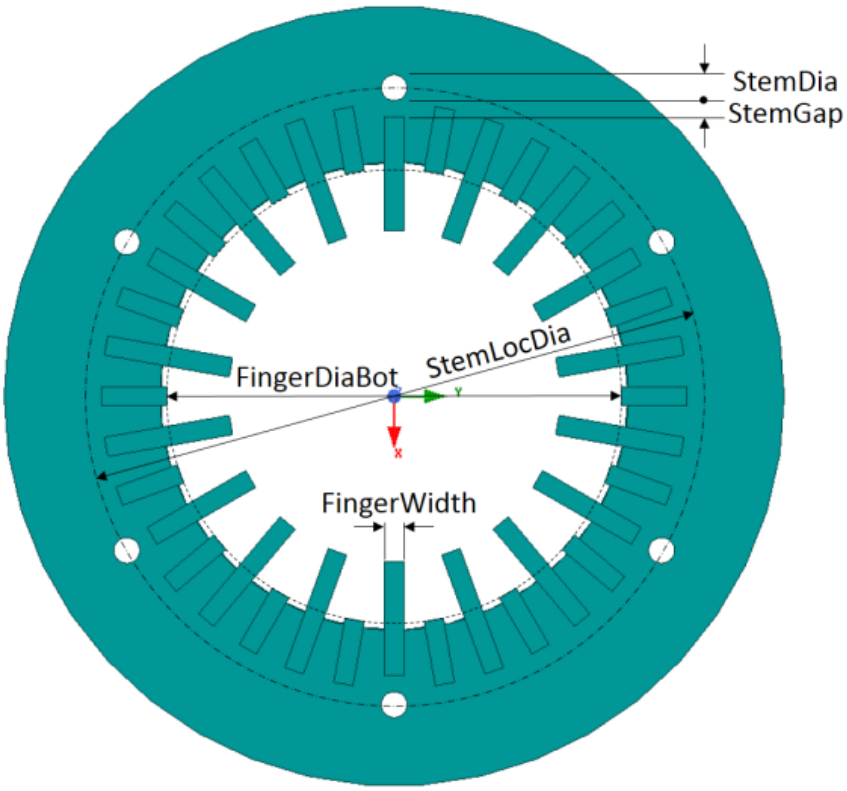
The PressPlate UDP is used to create press plates, fingers, and stems for large AC or DC machines. Press plates and fingers are put at two ends of a large laminated slot core and are locked in place by the stems so that the core laminations can be pressed tightly together.

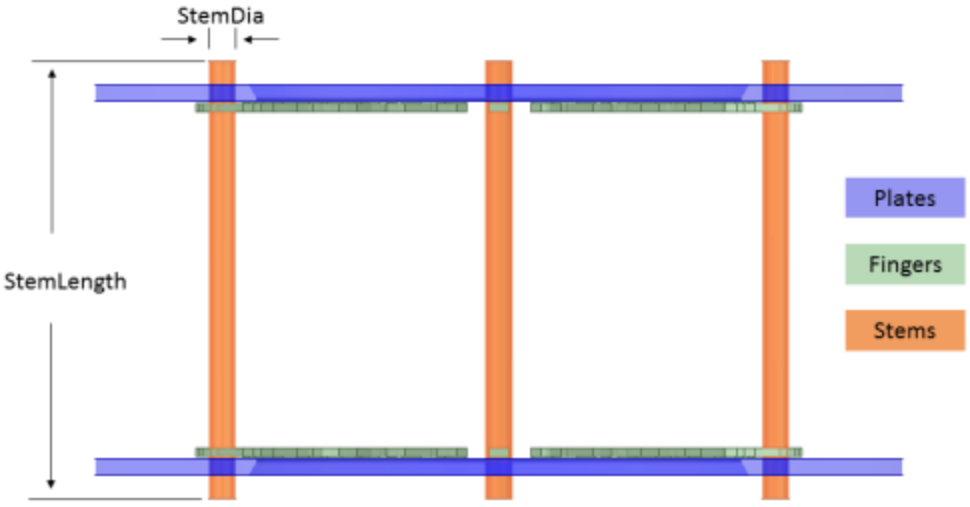
Property	Description
CoreLength	Axial length of the core inside press plates.
PlateDiaIn	Press plate inner diameter.
PlateDiaOut	Press plate outer diameter.
PlateThick	Press plate axial thickness.
PlateFilletThick	Plate fillet thickness at inner diameter.
PlateFilletAngle	Plate fillet angle at inner diameter.
Fingers	Number of end fingers, or the number of core slots.
FingerDiaIn	Inner diameter for distributed fingers.
FingerDiaOut	Outer diameter for distributed fingers.
FingerDiaHold	Finger holder outer diameter.
FingerDiaBot	Inner diameter for slot-bottom fingers, 0 without bottom fingers.
FingerWidth	Finger width to cover tooth width.
FingerThick	Finger holder axial thickness.
FingerPlateGap	Length of the axial gap between fingers and plates.
Stems	Number of stems to lock two end press plates.

Property	Description
StemLength	Axial stem length.
StemDia	Stem diameter.
StemLocDia	Diameter of stem center layout circle.
StemGap	Clearance gap around a stem.
InfoPart	Enter 0 to generate all parts (plates, fingers, and stems); 1 for press plates only; 2 for fingers; 3 for stems.

These parameters are shown in the following 2D examples:

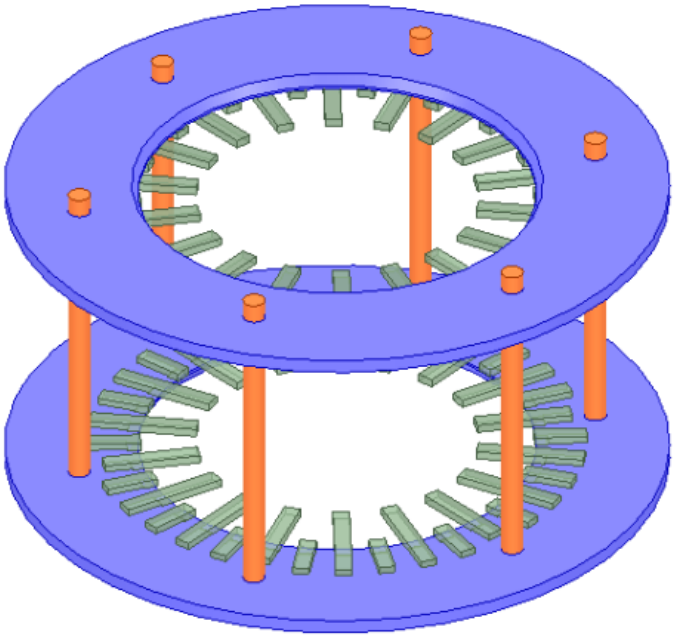






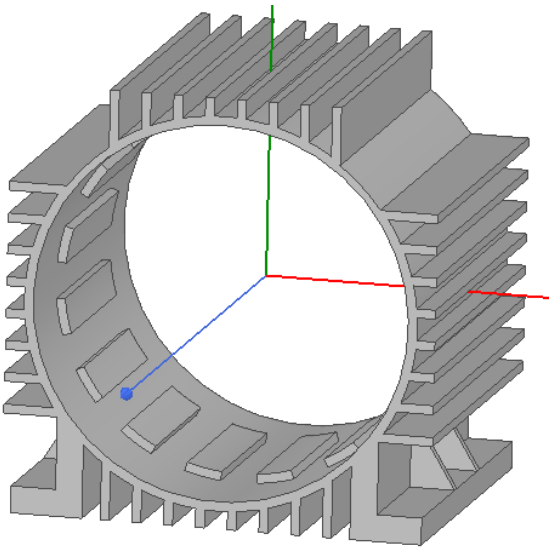
The following figure shows a typical arrangement of press plates, fingers, and stems in which their are 18 fingers under the press plates. The 6 stems serve to "lock" these parts at the two ends of the machine's laminated core (not shown), holding the entire assembly firmly together.

Press plates with fingers, and stems



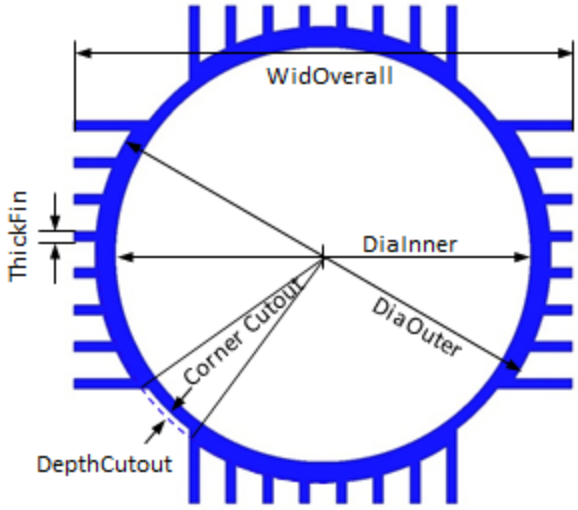
Housing UDP

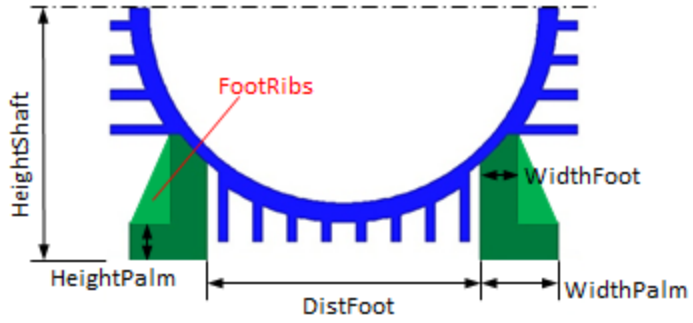
The Housing UDP is used to create the housing parts for machines. These include the fins, cooling ducts, and feet that support machines.

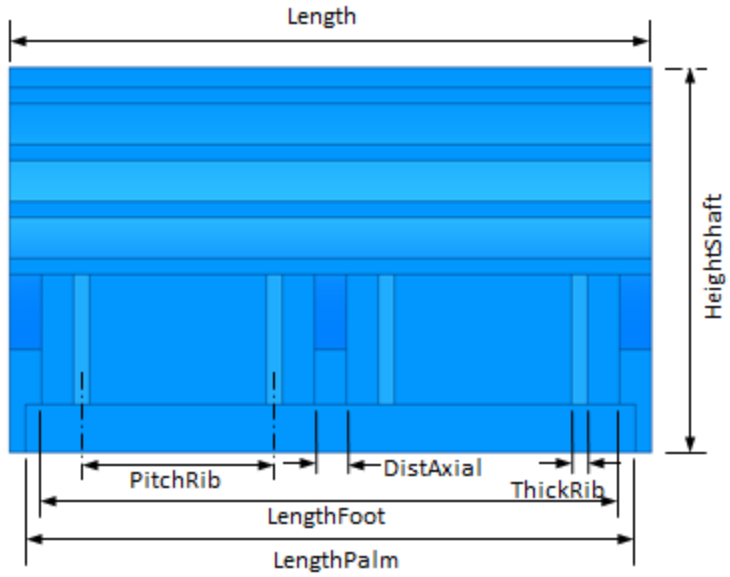


Property	Description
DiaOuter	Housing outer diameter
DiaInner	Housing inner diameter
WidOverall	Housing overall width
Length	Housing length
Fins	Number of fins; Fins = 0 for uniform distribution
FinType	1: radial round; 2: radial square; 3: parallel round; 4: parallel square; 5: squre ring; 6: servo ring
RatioCutout	Corner cutout ratio for FinType≠5
DepthCutout	Corner cutout depth for FinType≠5; min fin depth for FinType=5.
ThickFin	Fin thickness
Ducts	Number of ducts
DuctType	1: axial wind duct; 2: axial water jacket; 3: spiral water jacket (see below)
HeightDuct	Duct height in radial direction
EmbraceDuct	Ratio of the duct span, or duct width, to duct pitch. 0 < EmbraceDuct < 1
LengthDuct	Axial duct length for axial wind ducts only
SegAngle	Deviation angle for spiral water jacket; 0 for true-surface circular jacket
WidthFoot	Foot width at the junction with housing
LengthFoot	Foot end-to-end length in axial direction
DistFoot	Distance between two feet
DistAxial	Foot distance in axial direction
WidthPalm	Palm width at the junction with base plate
HeightPalm	Palm Height
LengthPalm	Palm length in axial direction
HeightShaft	Shaft-center Height from palm bottom
FootRibs	Number of reinforcing ribs for foot and palm
ThickRib	Reinforcing-rib thickness
PitchRib	Pitch between two reinforcing ribs
OffsetFoot	Axial offset of foot center to housing center (not currently used)
OffsetPalm	Axial offset of palm center to housing center (not currently used)
InfoPart	0: housing; 100: ducts.

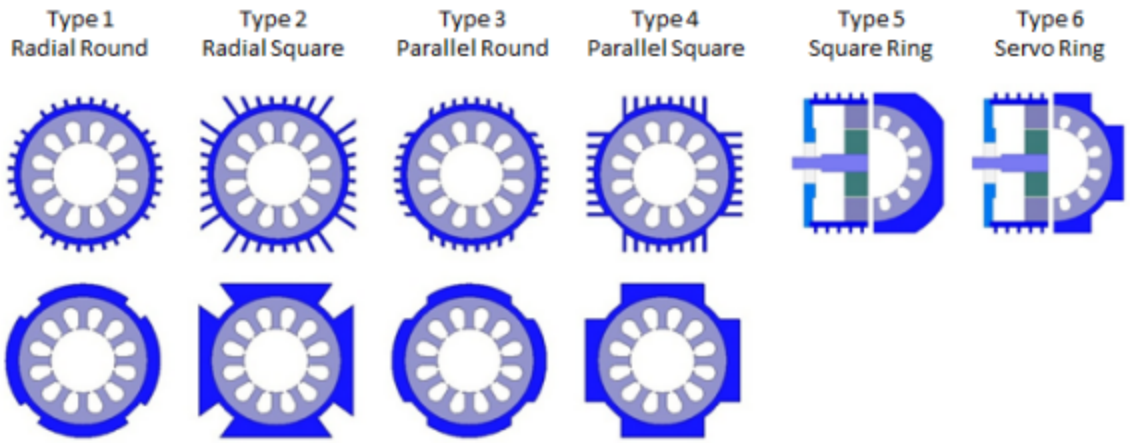
These parameters are shown in the following 2D examples:



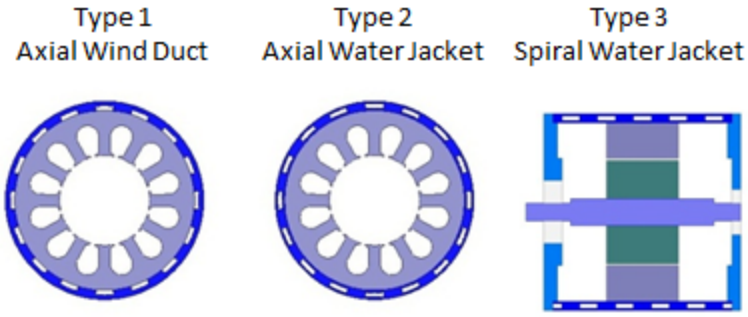




Housing FinType

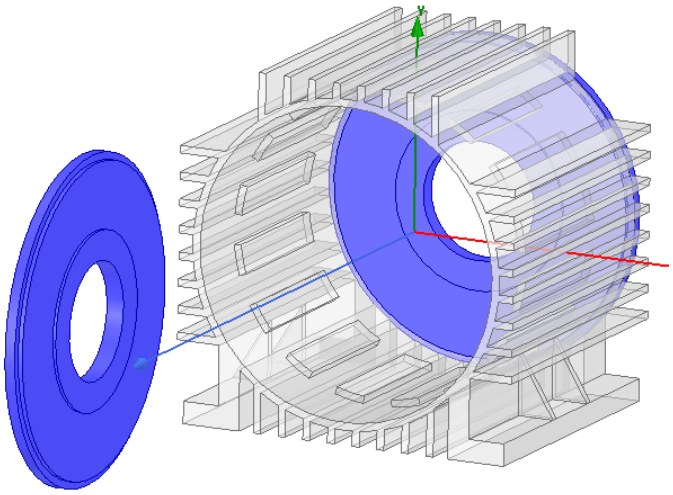


Housing DuctType



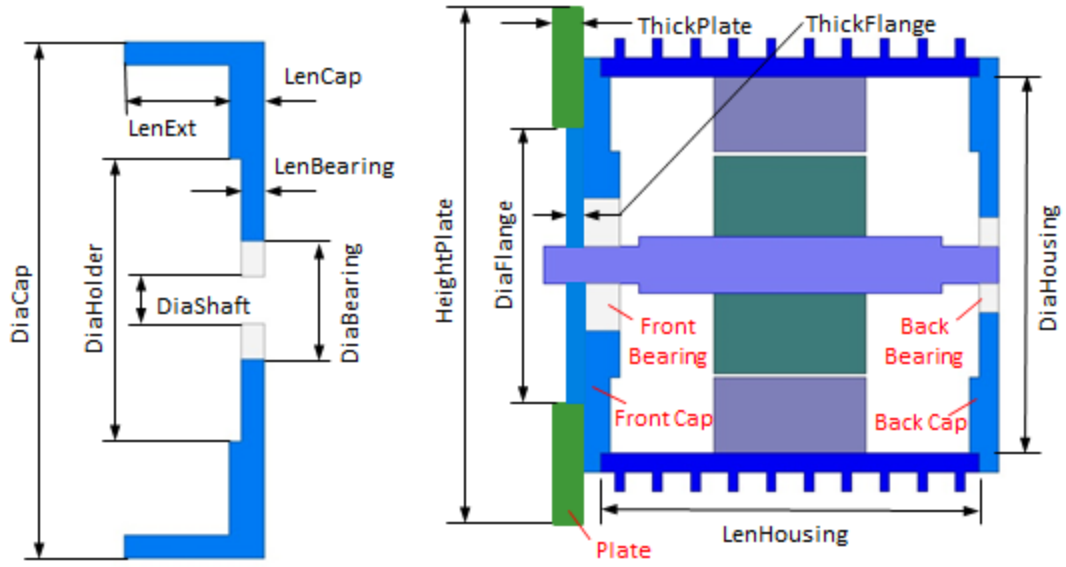
EndCap UDP

The EndCap UDP is to used to add the end caps, bearing surfaces, and rectangular cooling plate for a housing.



Property	Description
DiaHousing	Housing rabbet inner diameter
LenHousing	Housing axial length
DiaCap	End cap outer diameter
LenCap	Axial length, or thickness, of end cap
LenExt	Length from housing end to cap inner surface, <0 for rabbet length
DiaHolder	Outer diameter of bearing holder
DiaBearing	Bearing outer diameter
LenBearing	Bearing axial length
DiaShaft	Shaft diameter, or bearing inner diameter
DiaFlange	Diameter of the flange section that covers the front bearing
ThickFlange	Thickness of the flange section that covers the front bearing
HeightPlate	Height of the rectangular cooling plate
WidthPlate	Width of the rectangular cooling plate
ThickPlate	Thickness of the rectangular cooling plate
InfoPart	0: front cap; 1: back cap; 2: front bearing; 3: back bearing; 4: flange.

These parameters are shown in the following 2D examples:



Helix UDPs

Helix UDPs are shapes of inductance design and are widely used in electronic devices and signal integrity area.

Helix UDPs consist of the following types:

- PolygonHelix UDP
- RectHelix UDP

PolygonHelix UDP

The shape of the cross-section of a PolygonHelix is a polygon.

Property	Description
PolygonSegments	Number of cross-section polygon segments, 0 for circle.
PolygonRadius	Outer radius of cross-section polygon.
StartHelixRadius	Start radius from polygon center to helix center.
RadiusChange	Radius change per turn.
Pitch	Helix pitch.
Turns	Number of turns.
SegmentsPerTurn	Number of segments per turn, 0 for true surface.
RightHanded	Helix direction, non-zero for right handed.

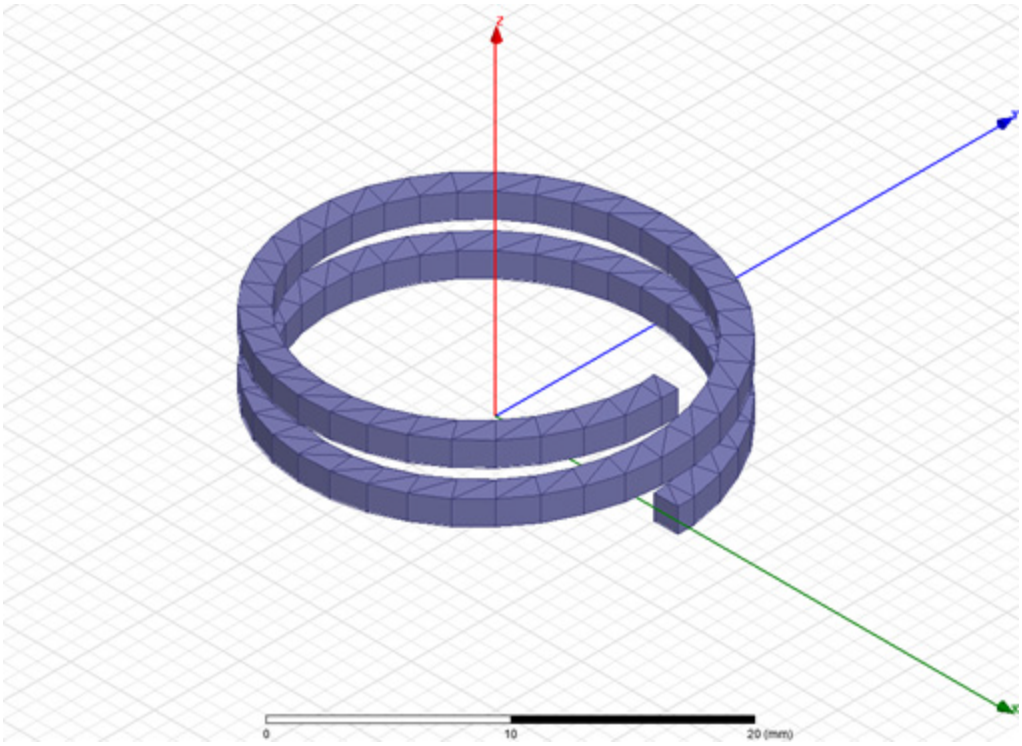


Figure 9-1 Overview of PolygonHelix UDP

Note	Figure 9-1 is an overview of the PolygonHelix UDP. “Polygon” means the cross-section of the helix is a polygon, which by default is a square (**PolygonSegments** set to 4). Polygon helices in Figure 9-2 and Figure 9-3 have the default settings of this UDP, whereas each of the polygon helices in Figure 9-4 to Figure 9-8 has only one parameter value that is different from the defaults shown in Figure 9-2 and Figure 9-3 .

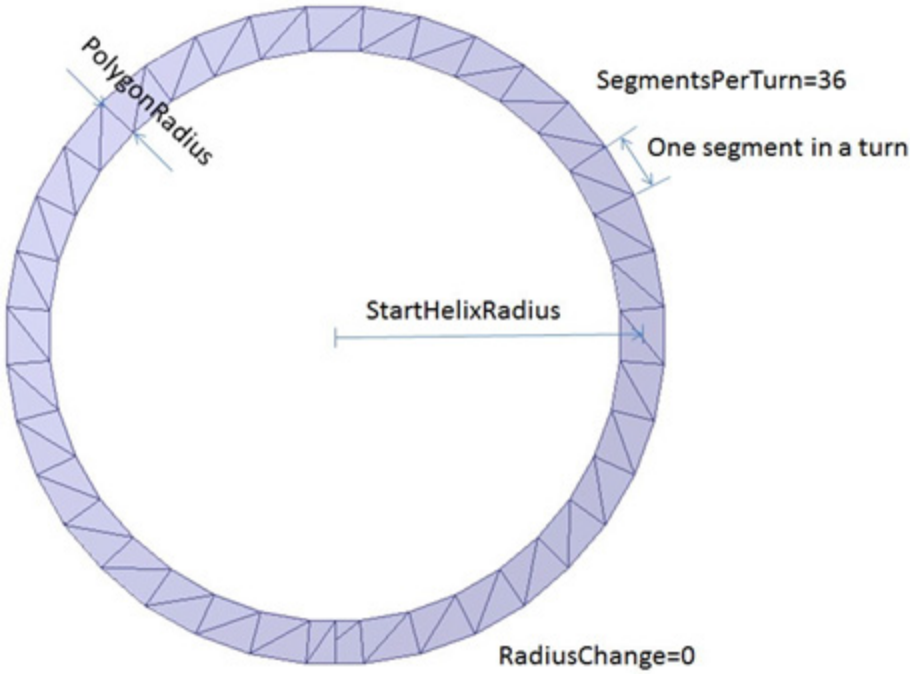


Figure 9-2 Parameters of PolygonHelix UDP on x-y plane.

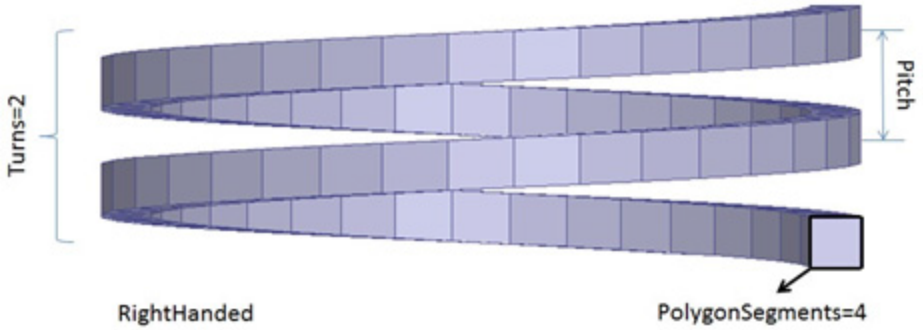


Figure 9-3 Parameters of PolygonHelix UDP on axial direction.

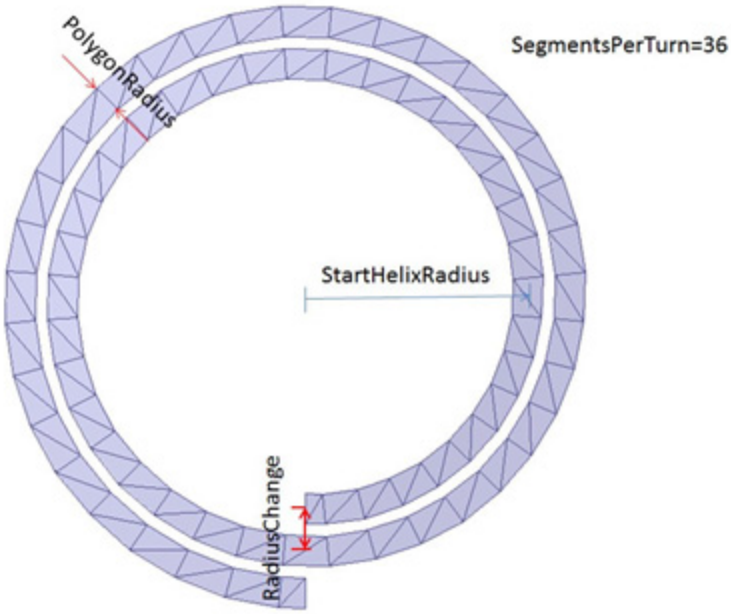


Figure 9-4 Different RadiusChange value (non-zero); the value in Figure 9-2 is 0.

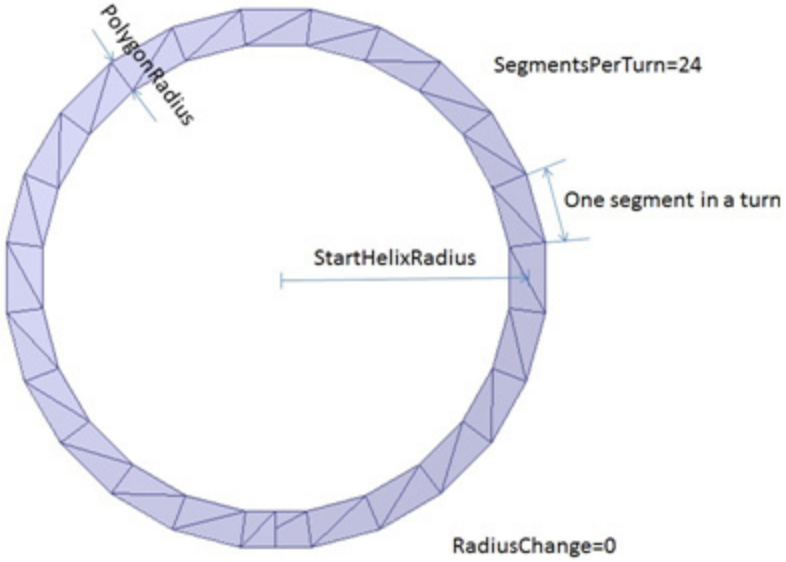


Figure 9-5 Different SegmentsPerTurn value (24); the value in Figure 9-2 is 36.

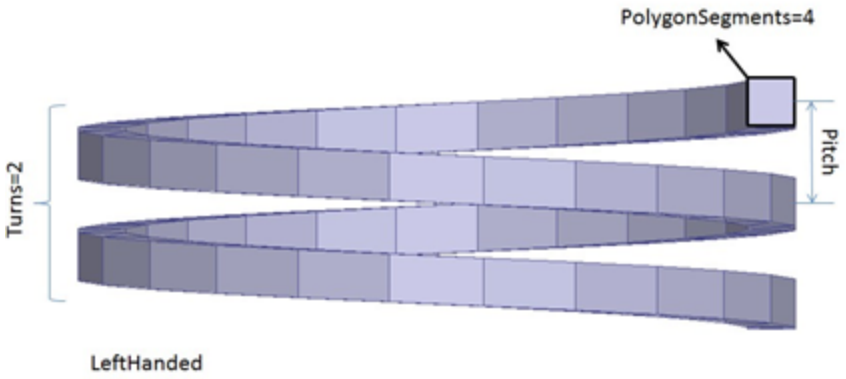


Figure 9-6 Left-handed (RightHanded set to 0) helix; Figure 9-3 shows a right-handed helix.

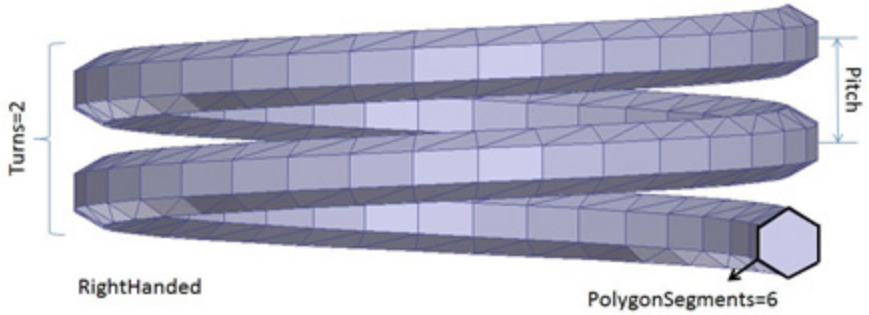


Figure 9-7 Different PolygonSegments value (6); the value in Figure 9-3 is 4.

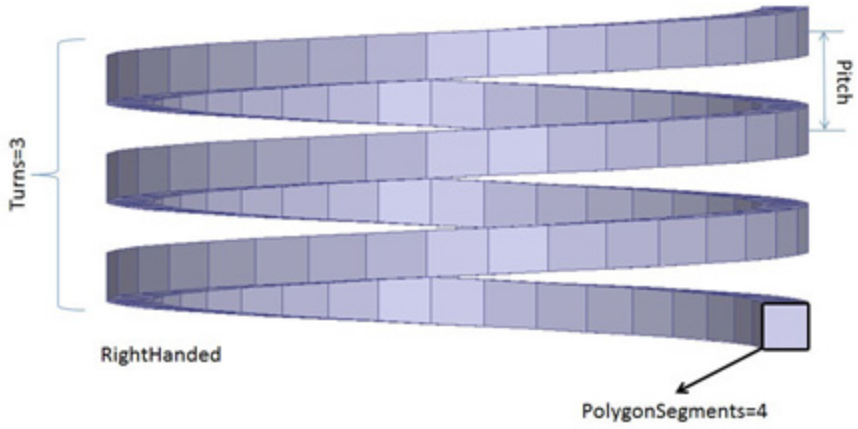


Figure 9-8 Different Turns value (3); the value inFigure 9-3 is 2.

Creating a Right-Handed or Left-Handed Polygon Helix

You can create a right-handed polygon helix manually by setting the value of **RightHanded** to 1 (or any non-zero value), which is set by default in Maxwell.

You can create a left-handed polygon helix by setting **RightHanded** to 0.

Examples of right- and left-handed polygon helices are shown in Figure 9-3 and Figure 9-6 , respectively.

You can assign the value of this property either when creating the UDP or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

RectHelix UDP

The shape of the cross-section in a RectHelix is a rectangle.

Property	Description
RectHeight	Cross-section rectangle height in z direction.
RectWidth	Cross-section rectangle width in r direction.
StartHelixRadius	Start radius from rectangle center to helix center.
RadiusChange	Radius change per turn.
Pitch	Helix pitch.
Turns	Number of turns.
SegmentsPerTurn	Number of segments per turn, 0 for true surface.
RightHanded	Helix direction, non-zero for right handed.

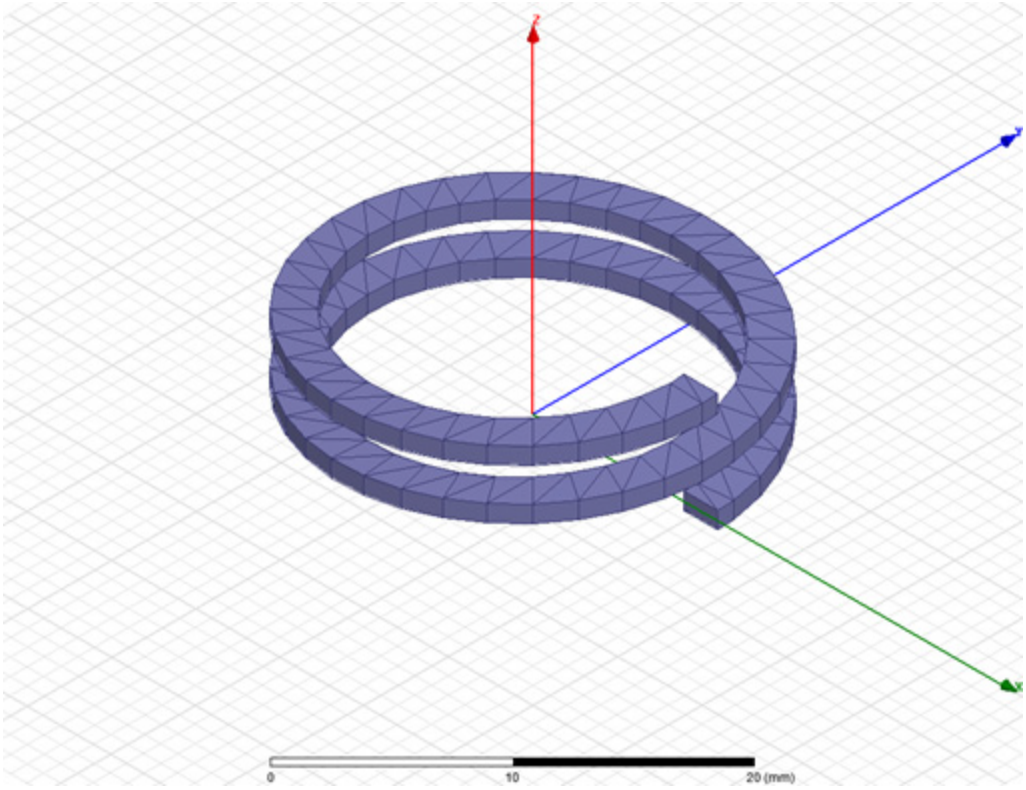


Figure 9-9 Overview of RectHelix UDP

Note	• The major difference between the RectHelix UDP and the PolygonHelix UDP is the shape of the cross-sections of the helices. In RectHelix, the shape of its cross-section is a rectangle whereas that of PolygonHelix is a polygon. • Except for **RectHeight** and **RectWidth**, parameters are the same as those in the Polygon Helix UDP. The one figure in this section describes **RectHeight** and **RectWidth**. • For *all other* parameters, refer to Polygon Helix UDP.

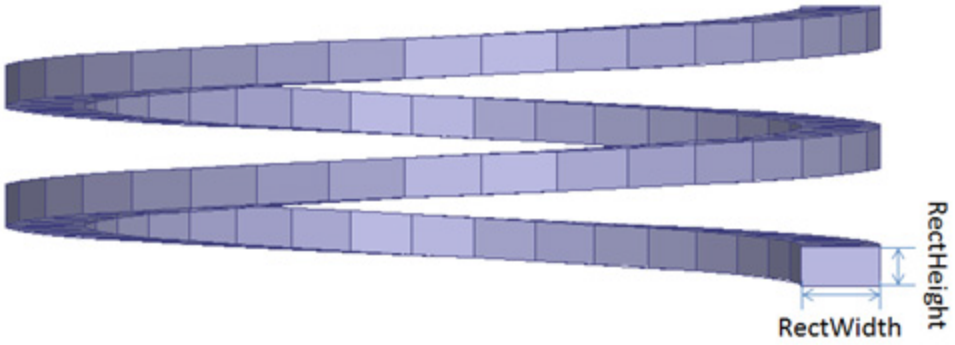


Figure 9-10 RectHeight and RectWidth in RectHelix UDP

Creating a Right-Handed or Left-Handed Rectangle Helix

You can create a right-handed rectangle helix manually by setting the value of **RightHanded** to 1 (or any non-zero value), which is set by default in Maxwell.

You can create a left-handed rectangle helix by setting **RightHanded** to 0.

Examples of right-handed and left-handed rectangle helices are shown in Figure 9-8 and Figure 9-11 , respectively.

You can assign the value of this property either when creating the UDP or after creating it, as described in Creating a Band and Creating an Outer Region, respectively.

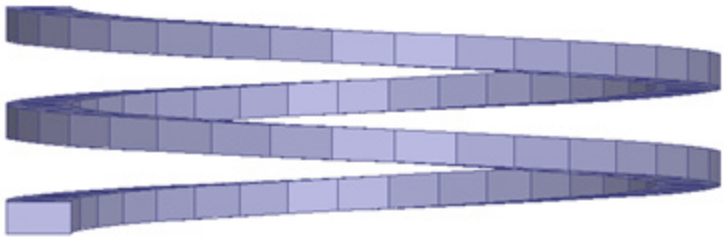


Figure 9-11 Left-handed rectangle helix

Information about Coding UDPs

This chapter uses an example of a Band UDP to provide information about how to write source code for a new UDP class when needed.

- Intrinsic Head Files presents some of the intrinsic structures and functions for all UDPs
- Code Implementation uses a Band UDP to illustrate what functions or procedures are needed to create a new UDP

Intrinsic Head Files

The following sections provide information on Intrinsic Structure Definitions and Intrinsic Function Declarations.

Head File for Intrinsic Structure Definition

Many intrinsic structures are written in C++ for UDPs. These intrinsic structures are provided in the UserDefinedPrimitiveStructures.h file (under directory: AnsysEM\Maxwell [x.x]\Win64\UserDefinedPrimitives\Examples\Headers).

Here are some common intrinsic structures that you need to know before starting coding for UDPs:

UDPPosition

Defines the most common and basic structure in a coordinate system, which is used to describe a point in the system.

```
struct UDPPosition
{
double x;
double y;
double z;
};
```

UDPPolylineDefinition

Defines the structure of polyline in a UDP, which is used to create sheets or lines in a UDP.

```
struct UDPPolylineDefinition
{
int noOfPoints;
struct UDPPosition* arrayOfPosition;
int noOfSegments;
struct UDPPolylineSegmentDefinition* arrayOfSegmentDefinition;
int isClosed;
int isCovered; /* Only if it is closed, then it can be covered
*/
```

```
};
```

UDPPrimitiveParameterDefinition

Defines the parameter/property definition of each UDP, such as *DiaGap*, *DiaYoke* and *Length* in the Properties window of each UDP. A definition for a parameter includes the name of the parameter, description of the parameter, unit of parameter, and default value of it.

```
struct UDPPrimitiveParameterDefinition
{
char* name; /* Name of the parameter */
char* description; /* some description of the parameter */
UDPUnitType unitType; /* No unit/length/angle */
double defaultValue;
};
```

UDPPolylineSegmentDefinition

Defines line segments used to form a polyline.

```
struct UDPPolylineSegmentDefinition
{
UDPPolylineSegmentType segmentType;
int segmentStartIndex;

/* Below information is required to be set only if you are
using Spline */
int numberOfPoints;

/* Below information is required to be set only if you are
using Angular Arc */
double angle; /* in degrees */
UDPPosition centerPoint;
UDPCoordinateSystemPlane arcPlane;
};
```

UDPSweepOptions

Sweep is a kind of operation to move an object from one position to another and meanwhile creating another object based on the track of the movement. For instance, sweep operations can be used to create a sheet from a line or a 3D object from a sheet. Different kinds of sweep types (and sweep angles) are used in SweepAlongAxis and SweepAroundAxis functions to make a sheet or a 3D object.

```
struct UDPSweepOptions
{
UDPSweepDraftType draftType;
```

```
double draftAngle;
double twistAngle;
};
```

UDPVector

Vector structures are used in UDP coding, mainly for rotation, moving or other kinds of transformations of geometry.

```
struct UDPVector
{
double x;
double y;
double z;
};
```

Head File for Intrinsic Function Declaration

Intrinsic function declarations are inside the same file (UserDefinedPrimitiveStructures.h) as intrinsic structure definitions. Functions are declared by 'typedef' keyword and function pointer. Some function declarations are shown below as examples:

Add a message to the message window, with desired severity level

This declaration is used in the DisplayInfo function in UDPs.

```
typedef void (*UDPAddMessageFP)
(
 UDPMessageSeverity, /* Message severity */
const char*, /* message string */
void* /* callbackData */
);
```

Function pointers used to create primitives

```
/* The return type for all primitives (except for spiral and
helix) is the ID of
 * the body created -1 is an invalid ID, which means the
operation failed.
 *
 * Spiral and helix are like sweep operations and operate on a
profile.
 * The return value indicates if the operation succeeded,
 * return value 0 means operation failed, any other value
 * means operation succeeded.
 */
```

```
/* 1D/2D ENTITIES */
typedef long (*UDPCreatePolylineFP)
(
const struct UDPPolylineDefinition* , /* polylineDefinition */
void* /* callbackData */
);

/* 3D ENTITIES */
typedef long (*UDPCreateCylinderFP)
(
UDPCoordinateSystemAxis, /* axis */
const struct UDPPosition*, /* center */
double, /* radius */
double, /* height */
void* /* callbackData */
);
```

Sweep functions

Sweep functions are used to create a sheet from a line or create a solid (3D object) from a sheet, as shown below.

```
  /* Sweep operations (Sweep along vector, around axis and
 * along path) modify the profile body. The resultant body
 * id is same as the profile id.
 *
 * In case of path sweep, the path body is consumed during the
 * sweep operation.
 *
 * The return value indicates
 * if the operation succeeded, return value 0 means operation
 * failed, any other value means operation succeeded.
 */
/* SweepAlongPath is used to sweep along a specific path to
create a sheet (from a line) or a solid (from a sheet)*/
typedef long (*UDPSweepAlongPathFP)
(
long, /* profileID */
long, /* pathID */
const struct UDPSweepOptions*, /* sweepOptions */
```

```
void* /* callbackData */
);

/* SweepAroundAxis is used to sweep around a specific axis (x,
y or z) to create a sheet (from a line) or a solid (from a
sheet)*/
typedef long (*UDPSweepAroundAxisFP)
(
long, /* profileID */
UDPCoordinateSystemAxis, /* sweepAxis */
double sweepAngle, /* sweepAngle */
const struct UDPSweepOptions*, /* sweepOptions */
void* /* callbackData */
);
```

Transform operations

The following two functions are used for transformations.

```
/* Transform operations (Translate i.e.move, rotate, mirror and
 * scale) modify the part body. The resultant body
 * id is same as the part id. The return value indicates
 * if the operation succeeded, return value 0 means operation
 * failed, any other value means operation succeeded.
 */

/* Translate function is used to do translational motion for a
given object. */
typedef long (*UDPTranslateFP)
(
long, /* partID */
const struct UDPVector*, /* translation vector */
void* /* callbackData */
);

/* Rotate function is used to rotate a given object. */
typedef long (*UDPRotateFP)
(
long, /* partID */
UDPCoordinateSystemAxis, /* axis */
double, /* angle */
```

```
void* /* callbackData */
);
```

Duplicate operation

```
/* Duplicate operations (Duplicate along line, around axis
 * and duplicate and mirror) create new bodies.
 *
 * In case of duplicate and mirror, only 1 duplicate can be
 * created. The ID of the duplicate body is returned by the
 * function. ID of -1 means the operation failed.
 *
 * For the other two cases multiple duplicates can be created.
 * The IDs of all such duplicates are returned in the function
 * parameter. The function return value indicates if the
operation
 * succeeded, return value 0 means operation failed, any other
 * value means operation succeeded.
 */
/* Duplicate an object around a given axis (of course can be
used to create either single or multiple copies around the
axis) */
typedef long (*UDPDuplicateAroundAxisFP)
(
long, /* partID */
UDPCoordinateSystemAxis, /* axis */
double, /* angle */
int, /* num of clones (including original) */
long*, /* returned clone ids */
void* /* callbackData */
);
```

Boolean operations

Boolean operations are used to unite, subtract, or intersect two objects.

```
/* Boolean operations (unite, subtract and intersect) modify
the
 * blank body. The resultant body id is same as the blank id.
 * Tool bodies are consumed during the operation.
 * The return value indicates if the operation succeeded,
 * return value 0 means operation failed, any other value
```

```
 * means operation succeeded.
 */
/* Unite two objects */
typedef long (*UDPUniteFP)
(
long*, /* objectIDs */
int, /* numObjects */
void* /* callbackData */
);

/* Subtract two objects */
typedef long (*UDPSubtractFP)
(
long*, /* blankIDs */
int, /* numBlanks */
long*, /* toolIDs */
int, /* numTools */
void* /* callbackData */
);
```

Function library with a collection of UDP functions

When coding for a UDP, you should declare an instance of UDPFunctionLib in the UDP and then use functions directly when needed.

For example:

```
UDPFunctionLib* functionLib;
functionLib->addMessage(kUDPInfoMessage, msg0, callbackData);

struct UDPFunctionLib
{
/* Utility functions */
UDPAddMessageFP addMessage;
UDPNameAFaceFP nameAFace;
UDPNameAEdgeFP nameAEdge;
UDPNameAVertexFP nameAVertex;
UDPGetFaceIDFromPositionFP getFaceIDFromPosition;
UDPGetEdgeIDFromPositionFP getEdgeIDFromPosition;
UDPGetTempDirPathFP getTempDirPath;
UDPGetInstallDirPathFP getInstallDirPath;
```

```
UDPGetSysLibDirPathFP getSysLibDirPath;
UDPGetUserLibDirPathFP getUserLibDirPath;
UDPGetPersonalLibDirPathFP getPersonalLibDirPath;
UDPSetProgressFP setProgress;

/* Primitives */
/* 1D/2D ENTITIES */
UDPCreatePolylineFP createPolyline;
UDPCreateRectangleFP createRectangle;
UDPCreateArcFP createArc;
UDPCreateCircleFP createCircle;
UDPCreateEllipseFP createEllipse;
UDPCreateRegularPolygonFP createRegularPolygon;
UDPCreateEquationBasedCurveFP createEquationBasedCurve;
UDPCreateEquationBasedSurfaceFP createEquationBasedSurface;
UDPCreateSpiralFP createSpiral;

UDPGetProjectPathFP getProjectPath;

/* BEGIN - RESERVED FOR FUTURE 1D/2D PRIMITIVES */
/* DO NOT INVOKE THESE RESERVED FUNCTIONS IN YOUR UDP DLL
CREATION CODE */
UDPReservedForFutureFP reserved1D2DPrimitive5;
UDPReservedForFutureFP reserved1D2DPrimitive6;
UDPReservedForFutureFP reserved1D2DPrimitive7;
UDPReservedForFutureFP reserved1D2DPrimitive8;
UDPReservedForFutureFP reserved1D2DPrimitive9;
UDPReservedForFutureFP reserved1D2DPrimitived10;
/* END - RESERVED FOR FUTURE 1D/2D PRIMITIVES */

/* 3D ENTITIES */
UDPCreateBoxFP createBox;
UDPCreateSphereFP createSphere;
UDPCreateCylinderFP createCylinder;
UDPCreateConeFP createCone;
UDPCreateTorusFP createTorus;
UDPCreatePolyhedronFP createPolyhedron;
```

```
UDPCreateHelixFP createHelix;

/* BEGIN - RESERVED FOR FUTURE 3D PRIMITIVES */
/* DO NOT INVOKE THESE RESERVED FUNCTIONS IN YOUR UDP DLL
CREATION CODE */
UDPReservedForFutureFP reserved3DPrimitive2;
UDPReservedForFutureFP reserved3DPrimitive3;
UDPReservedForFutureFP reserved3DPrimitive4;
UDPReservedForFutureFP reserved3DPrimitive5;
UDPReservedForFutureFP reserved3DPrimitive6;
UDPReservedForFutureFP reserved3DPrimitive7;
UDPReservedForFutureFP reserved3DPrimitive8;
UDPReservedForFutureFP reserved3DPrimitive9;
UDPReservedForFutureFP reserved3DPrimitive10;
/* END - RESERVED FOR FUTURE 3D PRIMITIVES */

/* Operations */
/* Booleans */
UDPUniteFP unite;
UDPSubtractFP subtract;
UDPIntersectFP intersect;

/* Sweeps */
UDPSweepAlongVectorFP sweepAlongVector;
UDPSweepAroundAxisFP sweepAroundAxis;
UDPSweepAlongPathFP sweepAlongPath;

/* Transformation */
UDPTranslateFP translate;
UDPRotateFP rotate;
UDPMirrorFP mirror;
UDPScaleFP scale;

/* Duplicate */
UDPDuplicateAlongLineFP duplicateAlongLine;
UDPDuplicateAroundAxisFP duplicateAroundAxis;
UDPDuplicateAndMirrorFP duplicateAndMirror;
```

```
/* Connect */
UDPConnectFP connect;

/* Offset */
UDPOffsetFP offset;

/* Section */
UDPSectionFP section;

/* Split */
UDPSplitFP split;

/* Import native body */
UDPImportNativeBodyFP importNativeBody;

/* Clone */
UDPCloneFP clone;

/* Delete part */
UDPDeletePartFP deletePart;

/* Create Object From Face */
UDPCreateObjectFromFaceFP createObjectFromFace;

/* Fillet/Chamfer */
UDPFilletFP fillet;
UDPChamferFP chamfer;

/* Detach Faces/Edges */
UDPDetachFacesFP detachFaces;
UDPDetachEdgesFP detachEdges;

UDPCreateObjectFromEdgeFP createObjectFromEdge;

UDPSheetThickenFP sheetThicken;
UDPSweepFaceAlongNormalFP sweepFaceAlongNormal;
```

```
/* cover/uncover */
UDPCoverLineFP coverLine;
UDPCoverSurfaceFP coverSurface;
UDPUncoverFacesFP uncoverFaces;

UDPSeparateBodiesFP separateBodies;

UDPMoveFacesFP moveFaces;
UDPWrapSheetFP wrapSheet;
UDPImprintFP imprint;
UDPImprintProjectionFP imprintProjection;

/* Import ansoft geometry */
UDPImportAnsoftGeometryFP importAnsoftGeometry;

/* correctly rerurns part ids */
UDPImportNativeBody2FP importNativeBody2;

UDPTransformFP transform;

/* BEGIN - RESERVED FOR FUTURE OPERATIONS */
/* DO NOT INVOKE THESE RESERVED FUNCTIONS IN YOUR UDP DLL
CREATION CODE */
UDPReservedForFutureFP reservedOperation20;
/* END - RESERVED FOR FUTURE OPERATIONS */
};
```

Code Implementation

This section provides several sample functions and structures to show how to write real code for a UDP. All codes are from a sample source file written for Band UDP.

There are two files, UDPBase.h and UDPBase.cpp files, that include basic structures and functions for all UDPs. All UDPs should be built based on these two files.

Define UDP Parameters and Default Values

A structure is needed for parameters and default values at the beginning of each UDP. In the following code, UDPPrimitiveParameterDefinition, an intrinsic structure of UDP shown in Head

File for Intrinsic Structure Definition, is used to create an array of parameters for a UDP.

```
struct UDPPrimitiveParameterDefinition primParams[] =
{
{"DiaGap", "Band diameter in gap center, DiaGap<DiaYoke for
outer band",
kUDPLengthUnit, 101},
{"DiaYoke", "Band diameter on yoke side, DiaYoke<DiaGap for
inner band",
kUDPLengthUnit, 40},
{"Length", "Band length", kUDPLengthUnit, 200},
{"SegAngle", "Deviation angle for band (0.1~5 degrees).",
kUDPAngleUnit, 3},
{"Fractions", "Number of circumferential fractions, 1 for
circular region.",
kUDPNoUnit, 1},
{"HalfAxial", "0: full model; 1: half model in axial
direction.", kUDPNoUnit, 0},
{"InfoCore", "0: band; 1: tool; 2: independent; 3: dependent;
100: region.", kUDPNoUnit, 0}
};
```

The function inside band UDP

```
int AreParameterValuesValid(char ** error, double* paramValues)
{
obj.IniParameters(paramValues, numOfParameters);
return obj.AreParametersValid(error);
}
```

IniParameters() and AreParametersValid() are two member functions, which are defined in the UDP base class (in UDPBase.cpp file) and shown below.

Initialize parameters and check for validity

First initialize parameters using IniParameters(), and then check the validity of values for parameters. These two functions are inside the UDPBase class and do not need to be implemented.

```
// set oriParams pointer, and make a copy of data to
paramValues
void CUDPBase::IniParameters(double* paramValues0, int num)
```

```
{
oriParams = paramValues0; // set oriParams pointer
numOfParams = num;
delete[] paramValues;
paramValues = new double [num];
for (int i=0; i<num; i++) paramValues[i] = oriParams[i]; //
make a copy
}
```

The following function is not required to be implemented, but it calls three other functions, GetParameters(), ValidateParameters and PutParameters(), which should be implemented in the new UDP class.

```
int CUDPBase::AreParametersValid(char ** error)
{
GetParameters(); // the implementation is shown below
checkOK = 0;
char *err = ValidateParameters(); // the implementation is
shown below
if (err) {
*error = err;
return 0;
}
PutParameters(); // the implementation is shown below
checkOK = 1;
return 1;
}
```

Parameter Validation Functions

This section covers the three functions that you need to implement in the new UDP classes. The functions could be used as templates for new UDP classes.

These functions are very straightforward. You should follow the examples to add or remove parameters for a new UDP.

GetParameters()

```
/* To get the values from array of parameters*/
void CBand::GetParameters()
{
diaGap = diaYoke = 0;
```

```
length = 0;
infoCore = 0;
infoOut = -1;

segAng = 3;
numFras = 1;
infoHalf = 0;

for (int i=0; i<numOfParams; i++) {
if (GetValue(i, "DiaGap", diaGap)) continue;
if (GetValue(i, "DiaYoke", diaYoke)) continue;
if (GetValue(i, "Length", length)) continue;

if (GetValue(i, "SegAngle", segAng)) continue;
if (GetValue(i, "Fractions", numFras)) continue;
if (GetValue(i, "HalfAxial", infoHalf)) continue;
if (GetValue(i, "InfoCore", infoCore)) continue;
}
rRegion = diaYoke / 2;
}
```

PutParameters()

```
/* To store the values into array of parameters*/
void CBand::PutParameters()
{
for (int i=0; i<numOfParams; i++) {
if (PutValue(i, "DiaGap", diaGap)) continue;
if (PutValue(i, "DiaYoke", diaYoke)) continue;
if (PutValue(i, "Length", length)) continue;

if (PutValue(i, "SegAngle", segAng)) continue;
if (PutValue(i, "Fractions", numFras)) continue;
if (PutValue(i, "InfoCore", infoCore)) continue;
}
}
```

ValidateParameters()

```
/* To check parameters one by one to see if their values are
valid */
char* CBand::ValidateParameters()
{
if (!paramValues) return 0;

if (diaYoke < 0) diaYoke = 0;
if (diaGap <= 0) return "DiaGap must be larger than 0.";
if (length < 0) length = 0; // length==0 for 2D geometry
infoOut = diaYoke>diaGap ? 1 : -1;

if (numFras < 1) numFras = 1;
if (segAng > 5) segAng = 5;
if (segAng <= 0) segAng = 0;
else if (segAng < 0.1) segAng = 0.1;
if (segAng > 0) {
numSegs = (int)(360/segAng + 0.5);
int remain = numSegs % numFras;
if (remain > 0) numSegs += numFras - remain;
segAng = 360.0 / numSegs;
}
else numSegs = 0;

if (infoCore < 100) {
if (infoCore < 0) infoCore = 0;
if (infoCore > 3) infoCore = 3;
}

if (infoCore >= 100) {
if (diaYoke <= 0) return "DiaYoke must be larger than 0.";
}
else if (infoCore >= 1) { // for tool; independent & dependent
if (diaYoke <= diaGap) return "DiaYoke must be larger than
DiaGap.";
}
```

```
return 0;
}
```

Implement Main Function to Create Primitive

In this part, functions are used to create the UDP. You need to write two functions for UDP creation.

- The first function is the main function to create this UDP. It calls two other functions, one of which is shown in part 2.

```
extern "C" DLLEXPORT
long CreatePrimitive(struct UDPFunctionLib* functionLib, void*
callbackData,
double* paramValues)
{
obj.SetParameterPointer(paramValues);
return obj.CreatePrimitive(functionLib, callbackData);
}
```

- The following function is used to create a UDP based on different **InfoCore** values. For instance, a region is created when **InfoCore** is equal to 100 while a tool is created when **InfoCore** is equal to 2.

  For functions to create specific object like region, tool and independent, refer to Source Code Files for more detail.

```
long CBand::CreatePrimitive(UDPFunctionLib* functionLib, void*
callbackData)
{
if (!checkOK) return 0;

// display infomation of modified parameters
SetFunctionLib(functionLib, callbackData);
DisplayInfo();

// create region
if (infoCore == 100) {
long region = CreateRegion();
return region==-1 ? 0 : 1;
}

// create tool
if (infoCore == 1) {
```

```
long tool = CreateTool();
return tool==-1 ? 0 : 1;
}

// create independent
if (infoCore == 2) {
long independent = CreateIndependent();
return independent==-1 ? 0 : 1;
}

// create dependent
if (infoCore == 3) {
long dependent = CreateIndependent(false);
return dependent==-1 ? 0 : 1;
}

// create band
long band = CreateBand();
return band==-1 ? 0 : 1;
}
```

Map Parameters between Different Versions

Parameters mapping function is used when the UDP version changes or some new parameters are added into a UDP.

- You should implement the following function MapParametersDefinitionVersions(). It calls two other functions IniParameters() and MapParameters(), which is in the UDPBase class.

```
extern "C" DLLEXPORT
int MapParametersDefinitionVersions(double** newParamValues,
char* oldVersion,
int oldNumberOfParams, struct UDPPrimitiveParameterDefinition*
oldParamDefinition)
{
obj.IniParameters(*newParamValues, numOfParameters);
obj.MapParameters(oldParamDefinition, oldNumberOfParams);// map
para by keywords

int verN = getVerNum(oldVersion);
```

```
if (verN < 200) {
obj.SetDefaultParameter(*newParamValues, "HalfAxial", 0);
}

return 1;
}
```

- The following function is the MapParameters() function, which has already been implemented in the UDPBase class.

```
// map parameter values of old version to new version based on
keywords,
// which is written in UDPBase class and does not need to be
implemented by user
void CUDPBase::MapParameters(UDPPrimitiveParameterDefinition*
params, int oldNum)
{
if (!primParams) return;
int i, j, j0=0;
for (i=0; i<numOfParams; i++) {
oriParams[i] = primParams[i].defaultValue;
if (j0<oldNum && strcmp(primParams[i].name, params
[j0].name)==0) {
oriParams[i] = params[j0++].defaultValue;
continue;
}
for (j=0; j<oldNum; j++) {
if (strcmp(primParams[i].name, params[j].name) == 0) {
oriParams[i] = params[j].defaultValue;
j0 = j+1;
break;;
}
}
}
}
```

Source Code Files

This section provides examples for the following:

- UDPBase.h
- UDPBase.cpp
- Band.h
- Band.cpp

UDPBase.h

```
// File: UDPBase.h

#if !defined (UDP_BASE_H)
#define UDP_BASE_H

#include <math.h>
#include <string.h>
#include <stdio.h>

#include "../Headers/UserDefinedPrimitiveDLLInclude.h"

//***************************************
// Basic functions
//***************************************
// get version number * 100
int getVerNum(char *ver);

// get the distance of two points
double getDistance(UDPPosition p0, UDPPosition p1);

// filter segments with length < minD from a polyline
void purifyPolyline(UDPPolylineDefinition& polylineDefinition,
double minD);

//***************************************
// Base class for any UDP
//***************************************
class CUDPBase
{
protected:
double minD;
```

```
int numOfParams;
double *oriParams;
double *paramValues;
bool checkOK; // 1: check OK; 0: check failed

UDPFunctionLib* functionLib;
void* callbackData;

protected:
bool GetValue(int i, char *s, double& v);
bool PutValue(int i, char *s, double& v);
bool GetValue(int i, char *s, int& v);
bool PutValue(int i, char *s, int& v);
bool GetValue(int i, char *s, bool& v);
bool PutValue(int i, char *s, bool& v);

virtual void GetParameters() = 0;
virtual void PutParameters() = 0;
virtual char *ValidateParameters() = 0; // return NULL if
initialize OK
void SetFunctionLib(UDPFunctionLib* functionLib0, void*
callbackData0);

// display messages
void DisplayInfo(char *msg=0);

// create lines
long CreateLine(UDPPosition startPos, UDPPosition endPos);
long CreatePolyLine(UDPPosition *pos, int numPos);
long CreateZAxisPath(double z0, double z1);

// create sheets
long CreateSheet(UDPPosition *pos, int numPos, int isSheet=1);

// create solids
long SweepAlongAxis(long obj, UDPCoordinateSystemAxis axis,
double v0, double v1);
```

```
long SweepAroundAxis(long obj, UDPCoordinateSystemAxis axis,
double angle);

// edit objects
long RotateAroundZVector(long obj, double angle, double x0=0,
double y0=0);
long DuplicateAroundAxis(long obj, UDPCoordinateSystemAxis
axis, int num,
double angle=0);
long AddObject(long objBase, long objAdd);
long SubObject(long objBase, long objSub);

public:
CUDPBase();
~CUDPBase() {delete[] paramValues;}

void IniParameters(double* paramValues0, int num);
void SetParameterPointer(double* paramValues0) {oriParams =
paramValues0;}
void MapParameters(UDPPrimitiveParameterDefinition* params, int
oldNum);
void SetDefaultParameter(double *returnParam, char *key, double
defVal);
int AreParametersValid(char ** error);

virtual long CreatePrimitive(UDPFunctionLib* functionLib, void*
callbackData) = 0;
};

#endif //if !defined (UDP_BASE_H)
```

UDPBase.cpp

```
#include <math.h>
#include <stdlib.h>

#include "../Headers/UDPBase.h"

// used to get parameter names in DisplayInfo() function
```

```
extern UDPPrimitiveParameterDefinition primParams[];

const double PI = 3.1415926535897932;

//**************************************
// Basic functions
//**************************************
int getVerNum(char *ver)
{
if (!ver) return 0;
double verD = 0;
sscanf(ver, "%lf", &verD);
return int(verD*100 + 0.5);
}

double getDistance(UDPPosition p0, UDPPosition p1)
{
double dx = p0.x - p1.x;
double dy = p0.y - p1.y;
double dz = p0.z - p1.z;

return sqrt(dx*dx + dy*dy + dz*dz);
}

// filter segments with length < minD from a polyline
void purifyPolyline(UDPPolylineDefinition& polylineDefinition,
double minD)
{
int noOfSegments = polylineDefinition.noOfSegments;
int noOfPoints = polylineDefinition.noOfPoints;
UDPPosition* pos = polylineDefinition.arrayOfPosition;
UDPPolylineSegmentDefinition* edge =
polylineDefinition.arrayOfSegmentDefinition;

int i0=0, j0=0, i, j;
for (j=0; j<noOfSegments; j++) {
int i1 = edge[j].segmentStartIndex;
```

```
int i2 = i1 + 1;
if (edge[j].segmentType == kUDPArcSegment) i2++;
double length = getDistance(pos[i1], pos[i2 % noOfPoints]);
if (length <= minD) continue;
if (j0 == j) {j0++; i0=i2; continue;}

edge[j0].segmentStartIndex = i0;
edge[j0++].segmentType = edge[j].segmentType;
for (i=i1+1; i<=i2; i++) pos[++i0] = pos[i % noOfPoints];
}
polylineDefinition.noOfPoints = i0;
polylineDefinition.noOfSegments = j0;
}

//***********************************************
// Member function definition for base class
//***********************************************
CUDPBase::CUDPBase() {
minD = 0;

paramValues = 0;
checkOK = 0;

functionLib = 0;
callbackData = 0;
}

bool CUDPBase::GetValue(int i, char *s, double& v)
{
if (strcmp(primParams[i].name, s) == 0) {
v = paramValues[i];
return 1;
}
return 0;
}

bool CUDPBase::PutValue(int i, char *s, double& v)
```

```
{
if (strcmp(primParams[i].name, s) == 0) {
paramValues[i] = v;
return 1;
}
return 0;
}

bool CUDPBase::GetValue(int i, char *s, int& v)
{
if (strcmp(primParams[i].name, s) == 0) {
v = (int)paramValues[i];
return 1;
}
return 0;
}

bool CUDPBase::PutValue(int i, char *s, int& v)
{
if (strcmp(primParams[i].name, s) == 0) {
paramValues[i] = v;
return 1;
}
return 0;
}

bool CUDPBase::GetValue(int i, char *s, bool& v)
{
if (strcmp(primParams[i].name, s) == 0) {
v = paramValues[i] > 0;
return 1;
}
return 0;
}

bool CUDPBase::PutValue(int i, char *s, bool& v)
{
```

```
if (strcmp(primParams[i].name, s) == 0) {
paramValues[i] = v ? 1.0 : 0;
return 1;
}
return 0;
}

// set oriParams pointer, and make a copy of data to
paramValues
void CUDPBase::IniParameters(double* paramValues0, int num)
{
oriParams = paramValues0;
numOfParams = num;
delete[] paramValues;
paramValues = new double [num];
for (int i=0; i<num; i++) paramValues[i] = oriParams[i];
}

// map parameter values of old version to new version based on
keywords
void CUDPBase::MapParameters(UDPPrimitiveParameterDefinition*
params, int oldNum)
{
if (!primParams) return;
int i, j, j0=0;
for (i=0; i<numOfParams; i++) {
oriParams[i] = primParams[i].defaultValue;
if (j0<oldNum && strcmp(primParams[i].name, params
[j0].name)==0) {
oriParams[i] = params[j0++].defaultValue;
continue;
}
for (j=0; j<oldNum; j++) {
if (strcmp(primParams[i].name, params[j].name) == 0) {
oriParams[i] = params[j].defaultValue;
j0 = j+1;
break;;
}
```

```
}
}
}

// set default values based the keyword of *key
void CUDPBase::SetDefaultParameter(double *returnParam, char
*key, double defVal)
{
for (int i=0; i<numOfParams; i++) {
if (strcmp(primParams[i].name, key) == 0) {
returnParam[i] = defVal;
return;
}
}
}

int CUDPBase::AreParametersValid(char ** error)
{
GetParameters();
checkOK = 0;
char *err = ValidateParameters();
if (err) {
*error = err;
return 0;
}
PutParameters();
checkOK = 1;

return 1;
}

void CUDPBase::SetFunctionLib(UDPFunctionLib* functionLib0,
void* callbackData0)
{
functionLib = functionLib0;
callbackData = callbackData0;
}
```

```
void CUDPBase::DisplayInfo(char *msg0)
{
if (msg0) {
functionLib->addMessage(kUDPInfoMessage, msg0, callbackData);
return;
}

char msg[BUFSIZ];
for (int i=0; i<numOfParams; i++) {
double err = primParams[i].unitType==kUDPLengthUnit ? minD :
1e-6;
if (fabs(paramValues[i] - oriParams[i]) <= err) continue;
sprintf(msg, "%s is modified from %g to %g",
primParams[i].name, oriParams[i], paramValues[i]);
functionLib->addMessage(kUDPInfoMessage, msg, callbackData);
}
}

// create a line from startPos to endPos
long CUDPBase::CreateLine(UDPPosition startPos, UDPPosition
endPos)
{
UDPPosition posArray[2] = {startPos, endPos};
UDPPolylineSegmentDefinition SegArray = {kUDPLineSegment, 0};
UDPPolylineDefinition lineDef = {2, posArray, 1, &SegArray, 0,
0};
return functionLib->createPolyline(&lineDef, callbackData);
}

// create a polyline
long CUDPBase::CreatePolyLine(UDPPosition *pos, int numPos)
{
if (numPos <= 1) return -1;

long IDs[2] = {-1, -1};
for (int i=0; i<numPos-1; i++) {
IDs[1] = CreateLine(pos[i], pos[i+1]);
```

```
IDs[0] = AddObject(IDs[0], IDs[1]);
}

return IDs[0];
}

// Create a path on z-axis symmetrically with given length for
sweeping along a path
long CUDPBase::CreateZAxisPath(double z0, double z1)
{
UDPPosition startPos = {0, 0, z0};
UDPPosition endPos = {0, 0, z1};
return CreateLine(startPos, endPos);
}

// create a cover sheet of straight line edges
long CUDPBase::CreateSheet(UDPPosition *pos, int numPos, int
isSheet)
{
if (isSheet == 0) return CreatePolyLine(pos, numPos);

int numSeg = isSheet==1 ? numPos : numPos-1;
UDPPolylineSegmentDefinition *seg = new
UDPPolylineSegmentDefinition [numSeg];
for (int i=0; i<numSeg; i++) {
seg[i].segmentType = kUDPLineSegment;
seg[i].segmentStartIndex = i;
}

UDPPolylineDefinition profile = {numPos, pos, numSeg, seg,
isSheet, isSheet};
purifyPolyline(profile, minD);
if (profile.noOfPoints < 3) return -1;
long sheet = functionLib->createPolyline(&profile,
callbackData);
delete[] seg;
return sheet;
}
```

```
// sweep along the specified axis within (v0, v1)
long CUDPBase::SweepAlongAxis(long obj, UDPCoordinateSystemAxis
 axis,
double v0, double v1)
{
if (obj == -1) return obj;
if (functionLib==0 || callbackData==0) return obj;
if (v0 == v1) return obj;

UDPPosition startPos = {0, 0, 0};
UDPPosition endPos = {0, 0, 0};
if (axis == kUDPXAxis) {
startPos.x = v0;
endPos.x = v1;
}
if (axis == kUDPYAxis) {
startPos.y = v0;
endPos.y = v1;
}
if (axis == kUDPZAxis) {
startPos.z = v0;
endPos.z = v1;
}

UDPSweepOptions sweepOptions = {kUDPRoundDraft, 0.0, 0.0};
long sweepPath = CreateLine(startPos, endPos);
if (sweepPath == -1) return obj;
functionLib->sweepAlongPath(obj, sweepPath, &sweepOptions,
callbackData);

return obj;
}

// sweep around the specified axis for ang degrees without
twist
long CUDPBase::SweepAroundAxis(long obj,
UDPCoordinateSystemAxis axis, double angle)
```

```
{
if (obj == -1) return obj;
if (functionLib==0 || callbackData==0) return obj;
if (angle == 0) return obj;

UDPSweepOptions sweepOptions = {kUDPRoundDraft, 0.0, 0.0};
functionLib->sweepAroundAxis(obj, axis, angle, &sweepOptions,
callbackData);

return obj;
}

// rotate around a vector through (x0, y0) in parallel with Z
axis
long CUDPBase::RotateAroundZVector(long obj, double angle,
double x0, double y0)
{
if (obj == -1) return obj;
angle -= floor(angle/360) * 360;
if (angle == 0) return obj;

UDPVector vector = {-x0, -y0, 0};
functionLib->translate(obj, &vector, callbackData);
functionLib->rotate(obj, kUDPZAxis, angle, callbackData);
UDPVector vector2 = {x0, y0, 0};
functionLib->translate(obj, &vector2, callbackData);
return obj;
}

// duplicate an object around given axis and return the united
part
long CUDPBase::DuplicateAroundAxis(long obj,
UDPCoordinateSystemAxis axis,
int num, double angle)
{
if (obj == -1) return obj;
if (num == 1) return obj;
```

```
long *IDs = new long [num];
if (angle==0) angle = 360.0 / num;
long bSuccess = functionLib->duplicateAroundAxis(obj, axis,
angle, num, IDs,
callbackData);

long id = IDs[0];
if (bSuccess == 1) {
IDs[num - 1] = obj;
bSuccess = functionLib->unite(IDs, num, callbackData);
}
delete[] IDs;

return bSuccess==0 ? obj : id;
}

// add objAdd to objBase and unite them together
long CUDPBase::AddObject(long objBase, long objAdd)
{
if (objAdd < 0) return objBase;
if (objBase < 0) return objAdd;

long IDs[] = {objBase, objAdd};
functionLib->unite(IDs, 2, callbackData);
return objBase;
}

// subject objSub from objBase
long CUDPBase::SubObject(long objBase, long objSub)
{
if (objSub < 0) return objBase;
if (objBase < 0) return -1;

long bSuccess = functionLib->subtract(&objBase, 1, &objSub, 1,
callbackData);
if (bSuccess == 0) return -1;
```

```
return objBase;
}
```

Band.h

```
// File: Cores.h

#if !defined (CORES_H)
#define CORES_H

#include "../Headers/UDPBase.h"

//**************************************
// Basic class for Band
//**************************************
class CBand : public CUDPBase
{
protected:
double diaGap; // diameter at the gap side
double diaYoke; // diameter at the yoke side
double length; // core length or model depth: length = 0 for 2D

int numSegs; // nmuber of segments of whole circular band
int numFras; // number of circumferential fractions
int infoHalf; // 1 for half model in axial direction
int infoCore; // information for drawing different parts of the
object
int infoOut; // information for an inner or outer core

double rRegion; // region radius
double segAng; // angle per segment

protected:
virtual void GetParameters();
virtual void PutParameters();
virtual char *ValidateParameters();

long SweepAlongZAxis(long obj);
```

```
long CreateBand();
long CreateTool();
long CreateMaster(bool isMaster=true);
long CreateRegion(double r0=0, int fras=0);

public:
virtual long CreatePrimitive(UDPFunctionLib* functionLib, void*
callbackData);
};

#endif //if !defined (CORES_H)
```

Band.cpp

```
#include "../Headers/Band.h"

const double PI = 3.1415926535897932;

static struct UDPPrimitiveTypeInfo primitiveInfo =
{
"Band",
"Create a band, shaft or outer region",
"Ansys Incorporation",
"06-01-2012",
"2.0"
};

extern "C" DLLEXPORT
struct UDPPrimitiveTypeInfo* GetPrimitiveTypeInfo()
{
return &primitiveInfo;
}

struct UDPPrimitiveParameterDefinition primParams[] =
{
{"DiaGap", "Band diameter in gap center, DiaGap<DiaYoke for
outer band",
```

```
kUDPLengthUnit, 101},
{"DiaYoke", "Band diameter on yoke side, DiaYoke<DiaGap for
inner band",
kUDPLengthUnit, 40},
{"Length", "Band length", kUDPLengthUnit, 200},

{"SegAngle", "Deviation angle for band (0.1~5 degrees).",
kUDPAngleUnit, 3},
{"Fractions", "Number of circumferential fractions, 1 for
circular region.",
kUDPNoUnit, 1},
{"HalfAxial", "0: full model; 1: half model in axial
direction.", kUDPNoUnit, 0},
{"InfoCore", "0: band; 1: tool; 2: independent; 3: dependent;
100: region.", kUDPNoUnit, 0}
};

static int numOfParameters = sizeof(primParams)/sizeof
(primParams[0]);

extern "C" DLLEXPORT
int GetPrimitiveParametersDefinition(struct
UDPPrimitiveParameterDefinition**
paramDefinition)
{
*paramDefinition = primParams;
return numOfParameters;
}

extern "C" DLLEXPORT
char* GetLengthParameterUnits()
{
return "mm";
}

static CBand obj;

// In case of error this function should return 0
```

```
extern "C" DLLEXPORT
int AreParameterValuesValid(char ** error, double* paramValues)
{
obj.IniParameters(paramValues, numOfParameters);
return obj.AreParametersValid(error);
}

extern "C" DLLEXPORT
int MapParametersDefinitionVersions(double** newParamValues,
char* oldVersion,
int oldNumberOfParams, struct UDPPrimitiveParameterDefinition*
oldParamDefinition)
{
obj.IniParameters(*newParamValues, numOfParameters);
obj.MapParameters(oldParamDefinition, oldNumberOfParams);// map
para by keywords

int verN = getVerNum(oldVersion);
if (verN < 200) {
obj.SetDefaultParameter(*newParamValues, "HalfAxial", 0);
}

return 1;
}

extern "C" DLLEXPORT
long CreatePrimitive(struct UDPFunctionLib* functionLib, void*
callbackData, double* paramValues)
{
obj.SetParameterPointer(paramValues);
return obj.CreatePrimitive(functionLib, callbackData);
}

//***************************************************
// Member function definition for class CBand
//***************************************************
void CBand::GetParameters()
{
```

```
diaGap = diaYoke = 0;
length = 0;
infoCore = 0;
infoOut = -1;

segAng = 3;
numFras = 1;
infoHalf = 0;

for (int i=0; i<numOfParams; i++) {
if (GetValue(i, "DiaGap", diaGap)) continue;
if (GetValue(i, "DiaYoke", diaYoke)) continue;
if (GetValue(i, "Length", length)) continue;

if (GetValue(i, "SegAngle", segAng)) continue;
if (GetValue(i, "Fractions", numFras)) continue;
if (GetValue(i, "HalfAxial", infoHalf)) continue;
if (GetValue(i, "InfoCore", infoCore)) continue;
}
rRegion = diaYoke / 2;
}

void CBand::PutParameters()
{
for (int i=0; i<numOfParams; i++) {
if (PutValue(i, "DiaGap", diaGap)) continue;
if (PutValue(i, "DiaYoke", diaYoke)) continue;
if (PutValue(i, "Length", length)) continue;

if (PutValue(i, "SegAngle", segAng)) continue;
if (PutValue(i, "Fractions", numFras)) continue;
if (PutValue(i, "InfoCore", infoCore)) continue;
}
}

char* CBand::ValidateParameters()
{
```

```
if (!paramValues) return 0;

if (diaYoke < 0) diaYoke = 0;
if (diaGap <= 0) return "DiaGap must be larger than 0.";
if (length < 0) length = 0; // length==0 for 2D geometry
infoOut = diaYoke>diaGap ? 1 : -1;

if (numFras < 1) numFras = 1;
if (segAng > 5) segAng = 5;
if (segAng <= 0) segAng = 0;
else if (segAng < 0.1) segAng = 0.1;
if (segAng > 0) {
numSegs = (int)(360/segAng + 0.5);
int remain = numSegs % numFras;
if (remain > 0) numSegs += numFras - remain;
segAng = 360.0 / numSegs;
}
else numSegs = 0;

if (infoCore < 100) {
if (infoCore < 0) infoCore = 0;
if (infoCore > 3) infoCore = 3;
}



if (infoCore >= 100) {
if (diaYoke <= 0) return "DiaYoke must be larger than 0.";
}
else if (infoCore >= 1) { // for tool; independent & dependent
if (diaYoke <= diaGap) return "DiaYoke must be larger than
DiaGap.";
}

return 0;
}

long CBand::SweepAlongZAxis(long obj)
```

```
{
if (obj == -1) return obj;
if (functionLib==0 || callbackData==0) return obj;
if (length == 0) return obj;

double z0 = infoHalf==0 ? -length/2 : 0;
UDPSweepOptions sweepOptions = {kUDPRoundDraft, 0.0, 0.0};
long sweepPath = CreateZAxisPath(z0, length/2);
if (sweepPath == -1) return obj;
functionLib->sweepAlongPath(obj, sweepPath, &sweepOptions,
callbackData);

return obj;
}

long CBand::CreateBand()
{
long band = -1;
double r0 = diaGap / 2;
if (numSegs == 0) band = CreateRegion(r0);
else {
double angle = 2*PI / numSegs;
int n = numSegs / numFras;
double x0 = r0 * cos(angle);
double y0 = r0 * sin(angle);

UDPPosition pos[] = { 0, 0, 0,
r0, 0, 0,
x0, y0, 0};
band = CreateSheet(pos, 3);
band = SweepAlongZAxis(band);
band = DuplicateAroundAxis(band, kUDPZAxis, n, angle*180/PI);
}
if (diaYoke <= diaGap) return band;

long region = CreateRegion();
return SubObject(region, band);
```

```
}

long CBand::CreateTool()
{
UDPPosition p0 = {0, 0, -length/2};
double r0 = 0.55 * diaYoke;
long tool = functionLib->createCylinder(kUDPZAxis, &p0, r0,
length, callbackData);
long region = CreateRegion();
return SubObject(tool, region);
}

// Create independent or dependent
long CBand::CreateIndependent(bool isIndependent)
{
double r0 = 0.5 * diaYoke;
UDPPosition startPos = {0, 0, 0};
UDPPosition endPos = {r0, 0, 0};
long independent = CreateLine(startPos, endPos);
if (!isIndependent) RotateAroundZVector(independent,
360.0/numFras);
return SweepAlongZAxis(independent);
}

long CBand::CreateRegion(double r0, int fras)
{
if (r0 == 0) r0 = diaYoke/2;
if (fras == 0) fras = numFras;
UDPPosition startPos = {0, 0, 0};
UDPPosition endPos = {r0, 0, 0};
long region = CreateLine(startPos, endPos);
region = SweepAlongZAxis(region);
return SweepAroundAxis(region, kUDPZAxis, 360.0/fras);
}

long CBand::CreatePrimitive(UDPFunctionLib* functionLib, void*
callbackData)
{
```

```
if (!checkOK) return 0;

// display infomation of modified parameters
SetFunctionLib(functionLib, callbackData);
DisplayInfo();

// create region
if (infoCore == 100) {
long region = CreateRegion();
return region==-1 ? 0 : 1;
}

// create tool
if (infoCore == 1) {
long tool = CreateTool();
return tool==-1 ? 0 : 1;
}

// create independent
if (infoCore == 2) {
long independent = CreateIndependent();
return independent==-1 ? 0 : 1;
}

// create dependent
if (infoCore == 3) {
long dependent = CreateIndependent(false);
return dependent==-1 ? 0 : 1;
}

// create band
long band = CreateBand();
return band==-1 ? 0 : 1;
}
```

10 - Assigning Materials

You can add, remove, and edit materials in two main ways:

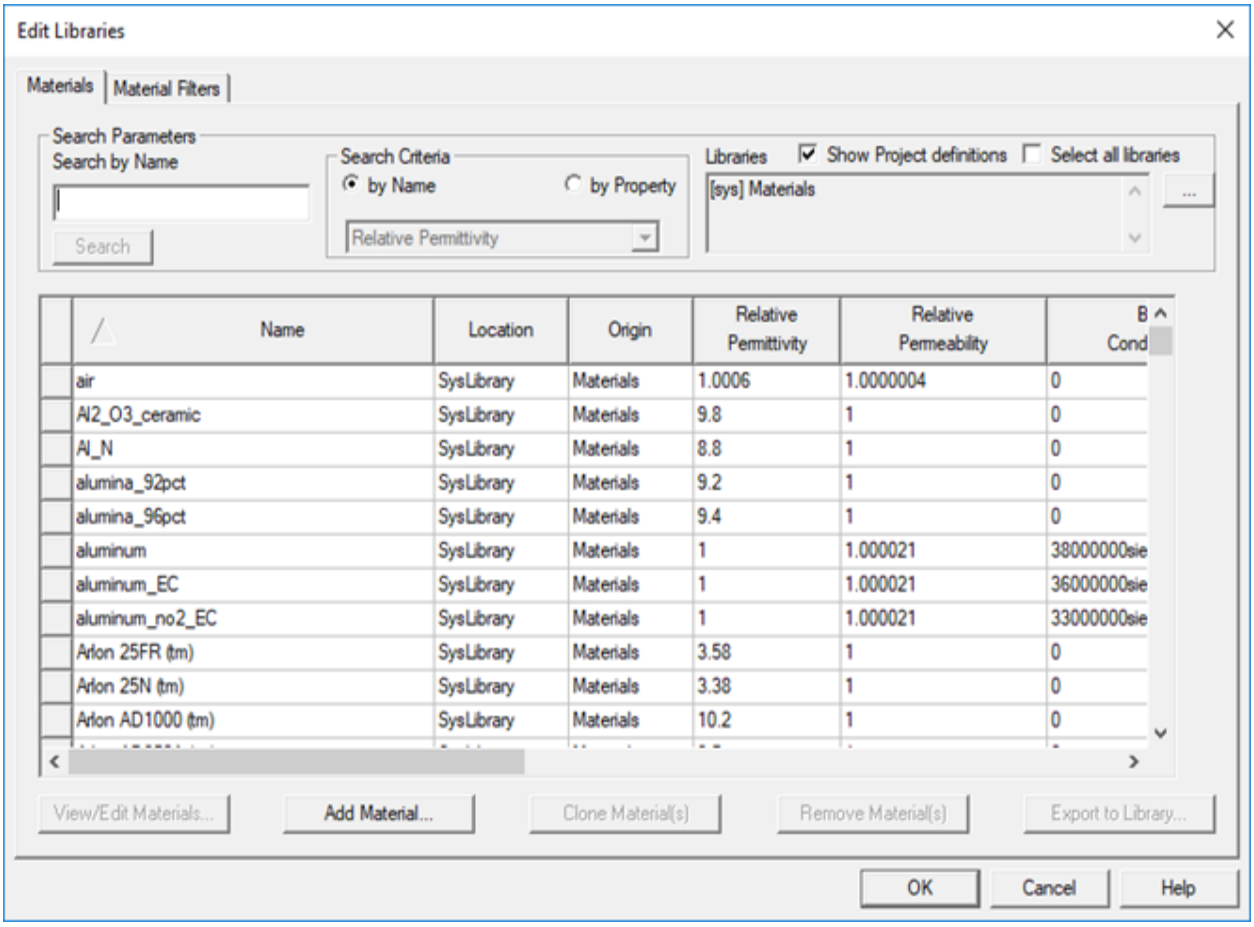
- Using the **Tools>Edit Libraries>Materials** menu command.
- Right-clicking **Materials** in the project tree and selecting **Edit Library**.

Regardless of which of the preceding two methods you use to edit a library, any new material you create exists only in the current project. Similarly, if you edit an existing material, the edited version becomes a *Project* material and exists only within the current project. To make a new or modified material available for other projects, you must export it to a user library and choose that library (or select **Show all libraries**) within the *Select Definition* dialog box.

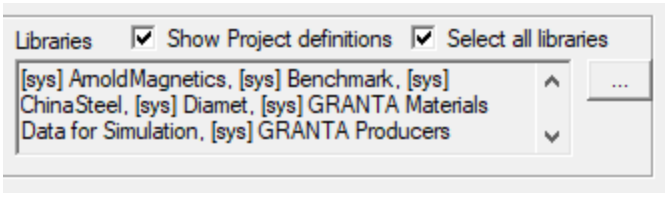
To assign a material to an object, follow the general procedure:

1. Select the object to which you want to assign a material.
2. Click **Modeler>Assign Material**  or select the Material field in the **Properties** window for the selected object, and select **Edit...** from the drop down menu.

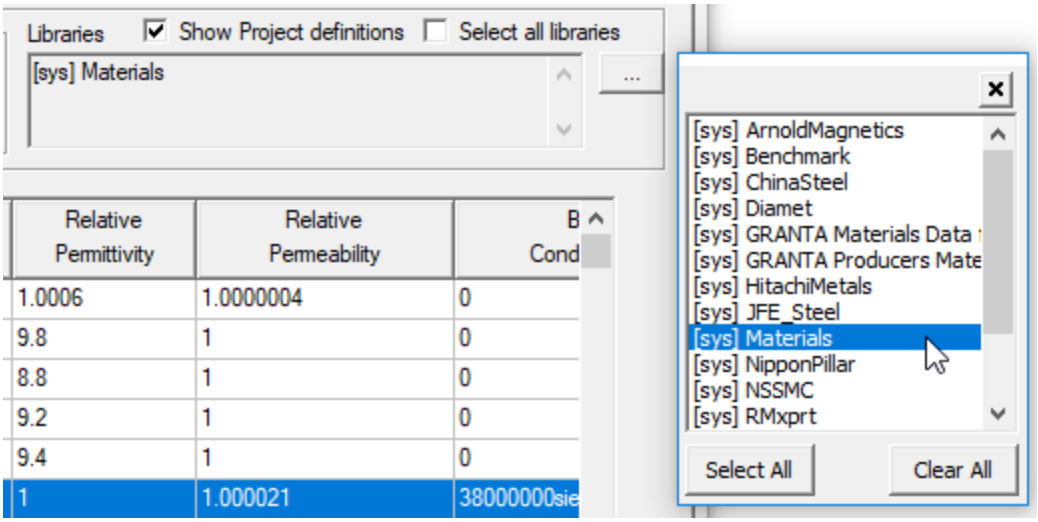
   The **Select Definition** window appears. The current material is highlighted, with the Name, Location, Origin library, and parameter values shown.



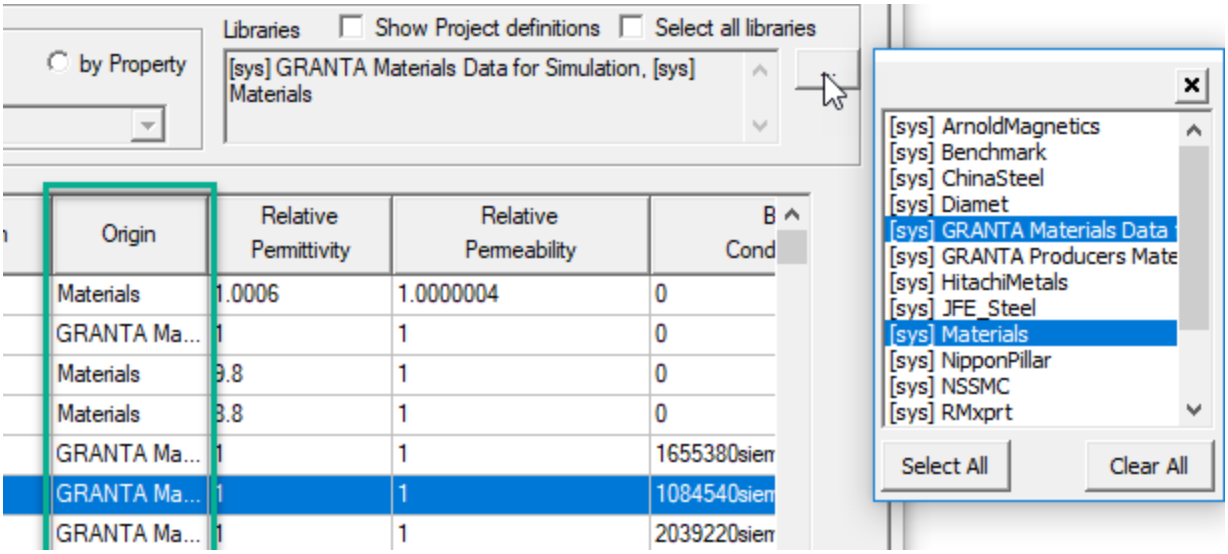
When the **Select all libraries** check box is selected, the window lists all of the materials in Ansys Electronics Desktop's global material library as well as the project's local material library.



If you click the ellipsis button **[...]**, you see a list of all available libraries. Any selected libraries are highlighted. Shift-Click allows you select a range. Ctrl-Click allows you to select any libraries.



The **Origin** column shows the originating library for each material, whether the **sys** library, or one of the additional libraries listed in the **Libraries** pane.



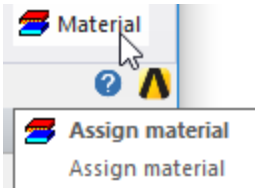
For further information on the materials and their intended uses, you can refer to the published information on materials from those libraries. For example, the Schott materials are described in detail in the Wily Series in Materials for Electronic & Optoelectronic Applications, *Microwave Dielectric Materials and Applications*, edited by M. T. Sebastian, Rick Ubic, and Heli Jantunen, volumes 1 and 2.

The GRANTA Materials Data for Simulation are an optional, licensed feature, including more than 700 generic materials and 500 producer-specific magnetic materials. When you use or view the GRANTA licensing libraries, the GRANTA license is checked out and held for 30 minutes. Viewing the contents of a GRANTA library in the Materials dialog or

modifying an object to use a GRANTA material causes a license check out and hold. If the license is already checked out, the hold time is reset to 30 minutes.

You can also open the **Select Definition** window in one of the following ways:

- In the **Properties** dialog box for the object, click the material name under the **Attributes** tab. A drop-down menu shows an **Edit...** button that opens the **Select Definition** window.
- With an object selected in the **Modeler** window, on the **Draw** ribbon, select the **Material** icon.



The menu also lists materials included in the current project. Selecting one of these materials provides another way to assign materials to an object.

- Right-click **Model** in the project tree, and then click **Assign Material** on the shortcut menu.
- Right-click the object in the history tree, and then click **Assign Material** on the shortcut menu.

3. Select a material from the list.

Note	You can search the listed materials by name or property value.

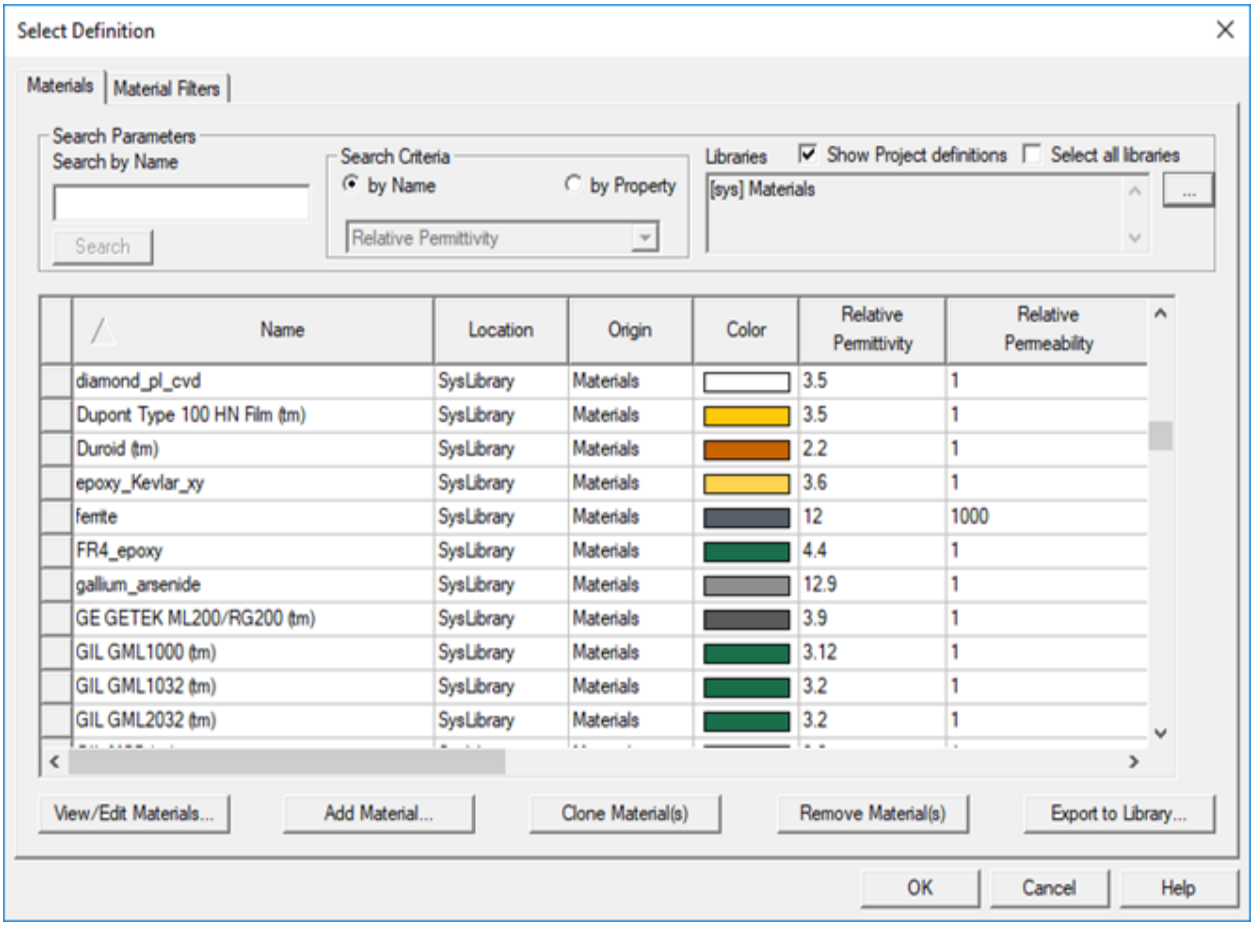
The list displays the currently set values for many of a material's properties such as Relative Permeability, Bulk Conductivity, and Composition. If the material you want to assign is not listed, you can add a new material to the global or local material library, and then select it.

4. Click **OK**.

The material you chose is assigned to the object.

Note	In the history tree, by default, objects are grouped by material. To change the default, select the object icon and right-click to display the **Group Objects by Material** check box.

If you have checked **Show Material Colors** on the Material Filters tab, the Materials tab includes a column showing a color swatch for each listed material.



You can edit the color and transparency values for materials in the View/Edit Material dialog.

Related Topics

Solve Inside or On a Surface
Adding New Materials
Removing Materials
Validating Materials
Sorting Materials
Viewing and Modifying Material Attributes
Copying Materials
Filtering Materials
Exporting Materials to a Library
Calculating Properties for Core Loss (BP Curve)
Calculating Properties for Temperature-dependent Core Loss Curves in Maxwell
Defining Variable Material Properties

Solving Inside or on the Surface

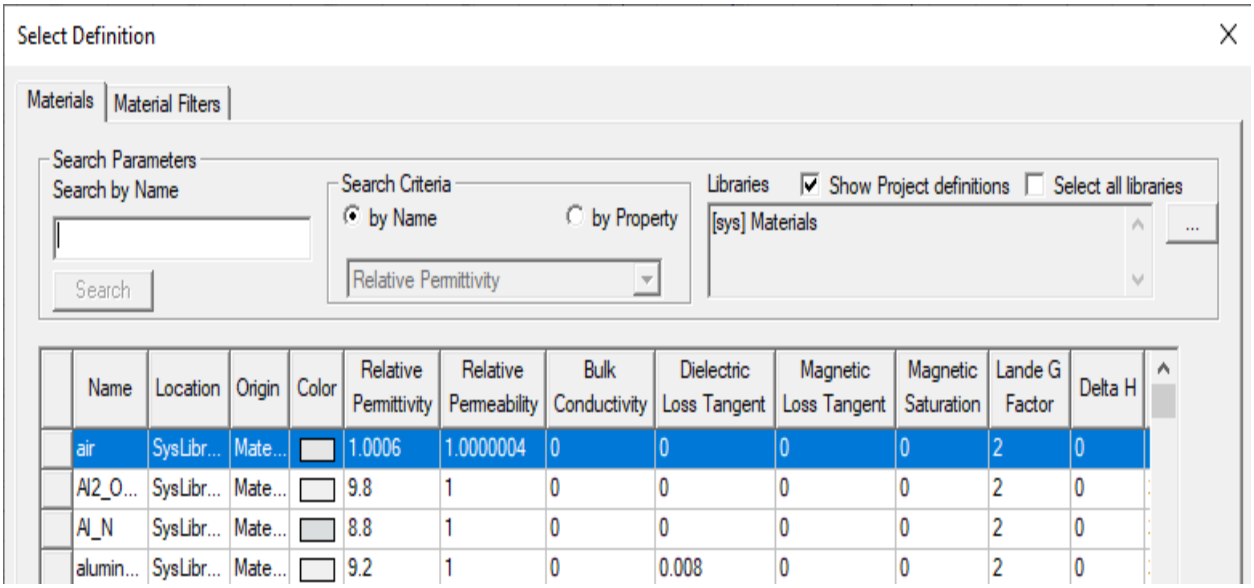
For Maxwell designs, when you assign a material to an object, you can specify whether to generate a field solution inside the object or on the surface of the object. If you elect to generate a solution inside the object, Maxwell will create a mesh inside the object and generate a solution from the mesh. If you elect to generate a solution on the surface of the object, Maxwell will create only a surface mesh for the object.

If you want a solution to be generated inside an object, select **Solve Inside** in the **Properties** window. Conversely, if you want a solution to only be generated on the surface of an object, clear the **Solve Inside** option in the **Properties** window.

By default, **Solve Inside** is selected for all objects with a bulk conductivity less than 10^5 siemens/meter and for perfect insulators. By default, the **Solve Inside** option in the **Properties** window is clear for perfect conductors.

Searching for Materials

You can search for materials in the **Select Definition** dialog box. The default Search Criteria is **by Name**, which is shown in the following example. Alternatively, you can choose to search by Property.

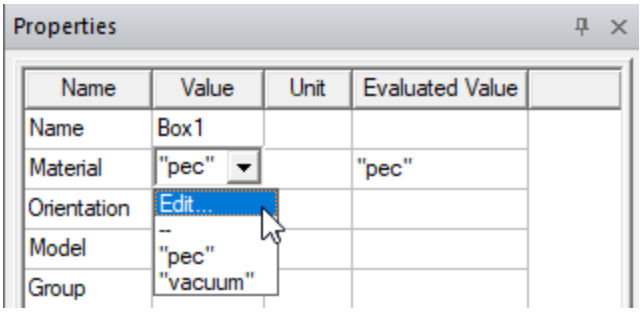


To search for a material **by Name**:

1. Access the **Select Definition** dialog box using one of the following methods:

   With one or more objects selected (that is, to assign a material to selected objects):

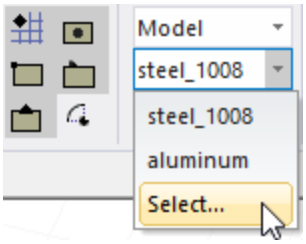
   - Click **Modeler** > **Assign Material**
   - Click the **Material***value* in the docked **Properties** window, and select **Edit...** from the drop-down menu.



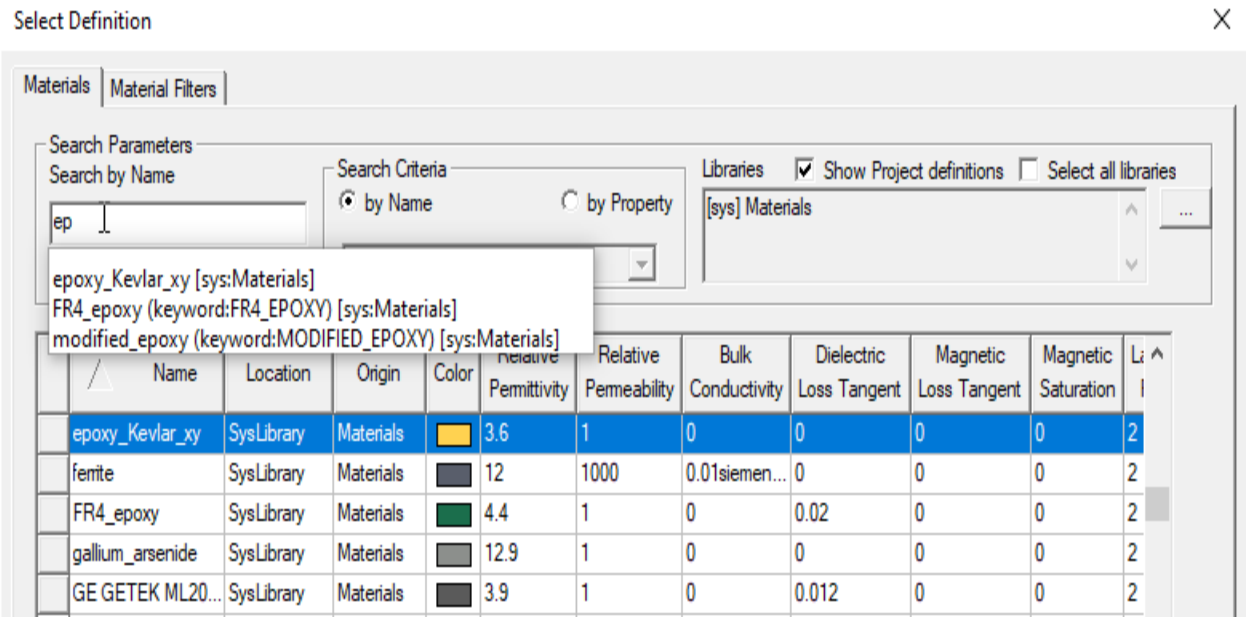
   - On the **Draw** ribbon tab, click **Assign material**.

   With nothing selected (that is, to set the default material):

- On the **Draw** ribbon tab, choose **Select** from the **Default material** drop-down menu:



2. In the **Search Criteria** section, ensure that **by Name** is selected, and in Libraries, specify the Libraries that you want to search. Only loaded libraries participate in text and keyword matching.
3. In the **Search by Name** text box, type a portion of the desired material name. The search text is case-insensitive.



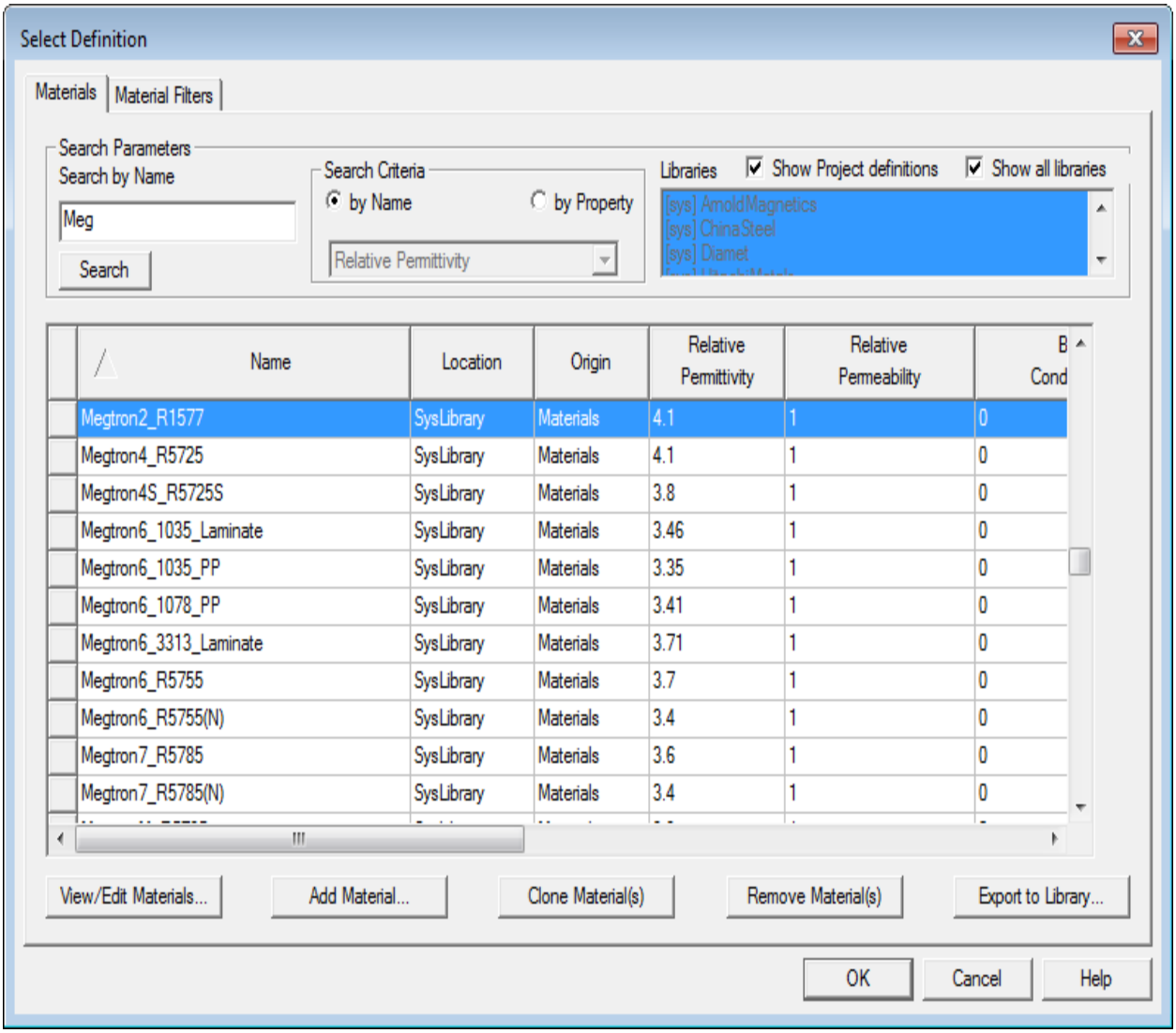
A drop down under the Search Parameters field will show the a list of the materials from selected libraries matching any part of the current text, and, if the Granta Materials library is selected, matching keywords in the material definitions. The first row containing the material name most similar to the characters you typed will be selected.

If the selected material is not the one you are searching for, do one of the following:

- Use the keyboard's arrow keys to select the material above or below the currently selected row.
- Use the scroll bar to scroll the listed materials upward or downward and click the desired material when it is visible.
- Type different characters in the **Search by Name** text box.

.

When the **Select all libraries** option is selected, the window lists all of the materials in Ansys Electronics Desktop's global material libraries that are applicable to the current design type as well as those in the project's local materials library. The Granta Materials Data for Simulation library now includes keywords that are used in the auto-complete matching. For example, entering "glass" will generate an auto-complete list of all materials with a name containing "glass" or with a keyword that contains "glass", all case-insensitive, for materials showing in the grid. The drop down of potential matches under the search field applies to all libraries. What shows in the grid is defined by which libraries are selected, what material filters are selected, project/design-specific validation, and whether project materials are selected for display.



Searching by Material Property

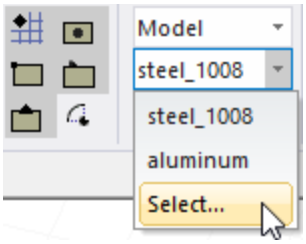
1. Access the **Select Definition** dialog box using one of the following methods:

   With one or more objects selected (that is, to assign a material to selected objects):

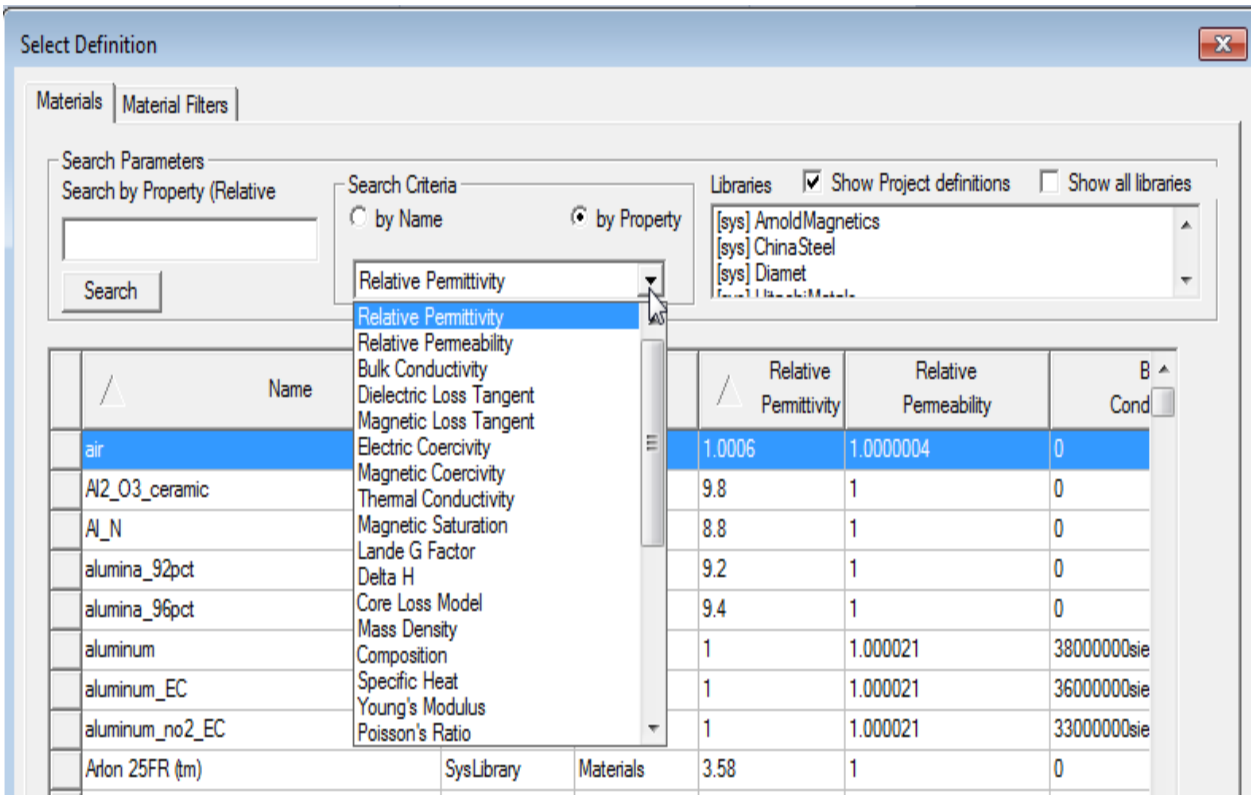
   - Click **Modeler** > **Assign Material**
   - Click the **Material***Value* in the docked *Properties* window, and select **Edit...** from the drop-down menu.
   - On the **Draw** ribbon tab, click **Assign material**.
   - On the **Draw** ribbon tab, click **Assign surface material**.

   With nothing selected (that is, to set the default material):

   - On the **Draw** ribbon tab, choose **Select** from the **Default material** drop-down menu:



2. In the **Search Criteria** area of the **Select Definition** dialog box, select **by Property**.



3. Select a material property from the pull-down list.

Note	By default, not all of the available properties are displayed in the materials table. Only the properties commonly used by the product are displayed. To view the complete table of properties, see Filtering Materials.

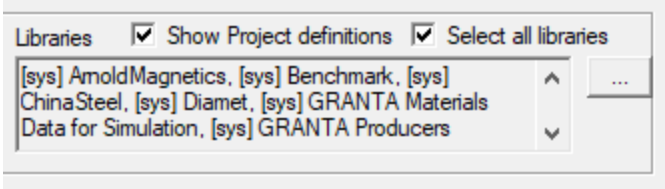
4. In the **Search Parameters** area, type a value for the property in the **Search by Property** box, and then click **Search**.

   The materials are sorted according to the value you entered. The material with the property value closest to the one you typed is selected.

   If the selected material is not the one you are searching for, do one of the following:

   - Use the keyboard's arrow keys to select the material above or below the currently selected row.
   - Use the scroll bar to scroll the listed materials upward or downward and click the desired material when it is visible.
   - Type a different numerical in the **Search by Property** text box and click **Search** again.

   When the **Show all libraries** check box is selected, the window lists all of the materials in Ansys Electronics Desktop's global material library as well as the project's local material library.



Adding New Materials

You can add a new material to a project or to the global user-defined material library. To make the new project material available to all projects, you must export the material to a global user-defined material library.

Materials are added using the **View/Edit Material** dialog box, which can be opened from either the **Select Definition** dialog box or the **Edit Libraries** dialog box.

To open the **Select Definition** dialog box:

- Click **Modeler>Assign Material**.

  The **Select Definition** dialog box appears.

To open the **Edit Libraries** dialog box:

- Click **Tools>Edit Libraries>Materials**. (In the project tree, you can also right-click **Materials**, and select **Edit Library**.)

  The **Edit Libraries** dialog box appears.

To add a new material:

1. From either the **Select Definition** dialog box or the **Edit Libraries** dialog box, click **Add Material**.

   The **View/Edit Material** dialog box appears.
2. Type a name for the material in the **Material Name** text box, or accept the default.
3. Select one of the following from the **Material Coordinate System Type** pull-down list:
   - **Cartesian** (default)
   - **Cylindrical**
   - **Spherical**
4. Under **View/Edit Material for** select:
   - **Active Design** to display properties used in the active design.
   - **Active Project** to display properties used in the active project.
   - **All Properties** to display all properties available. This enlarges the table of properties to show all properties possible for the selected Physics type(s). You can use the scroll bars or size the dialog to see all properties.

     Selecting **All Properties** also enables the three **Physics** check boxes: **Electromagnetic**, **Thermal**, and **Structural**. These check boxes are used to filter the properties displayed in the table by physics type. Clearing a check box hides properties of that physics type.
5. You can also enable the **View/ Edit Modifier** check box for **Thermal Modifier**.

   Checking this box causes the Thermal Column to display at the right side of the Properties of the Material table. Selecting **Edit** rather than **None** causes display of the Edit Thermal Modifier dialog.
6. Type a new name for the material in the **Material Name** text box or accept the default.
7. Select a material property type – **Simple** (i.e., *linear*), **Anisotropic**, **Nonlinear**, **Vector**, or **Custom** – for each property from the **Type** pull-down list. For Composition, values are **Solid**, **Lamination**, or **Litz Wire**. Some properties only use the Simple type. For thermal materials, select **Solid** or **Fluid** for the **Thermal Material Type**.

   Available selections vary with the property and solution type, and with the **View/Edit** selection as described above.

   For example, for an Eddy Current active design, if the material is **Simple** (i.e., *linear*), enter values for the following material properties in the **Value** boxes:
   - Relative Permeability (See also *Relative Permeability* in the Technical Notes.)
   - Relative Permittivity
   - Bulk Conductivity
   - Dielectric Loss Tangent
   - Magnetic Loss Tangent
   - Core Loss Model

- Composition: **Solid**, **Lamination**, or **Litz Wire**.

Note	The **Litz Wire** selection that appears under Composition is valid only for Maxwell 2D/3D Transient, and 2D/3D Eddy Current designs. Selecting **Litz Wire** for any other design type will generate an error message.

- Mass Density

If the material is a *ferrite*, enter a value greater than 0 in the Magnetic Saturation**Value** box. You may also choose to enter values in the Lande G Factor and Delta H**Value** boxes. Because Delta H values are measured at specific frequencies, you should also enter a - Measured Frequency value (default 9.4 Ghz).

Note	You may enter a variable name or mathematical expression in the **Value** box.

9. To modify the units for a material property, click the **Units** box, and then select a new unit system.
10. For **Material Appearance**, you can check the **Use Material Appearance** box to enable the fields for you to specify a color and transparency for the material.
    - Clicking the **Color** bar opens a standard color selection window.
    - Clicking the **Transparency** box opens a **Transparency** dialog with a text field and slider bar for selection.
11. If you want to add descriptive notes for the new material, click the ellipsis button [**...**] by the **Notes** field. This opens a dialog in which you can enter text. To enter multiple lines of notes, use **CTRL-Enter** to start a new line. (**Enter** by itself will save the Notes field and close the dialog.)
12. Below the list, in the **Calculate Properties for** pull-down list, do the following if your project contains a permanent magnet:
    a. Select **Permanent Magnet**.
    The **Properties for Permanent Magnet** dialog box appears.
    b. Select which parameters you want to define (**Mu**, **Hc**, **Br**, **Mp**).
    c. Type values in the text boxes, and select the units.
    d. Click **OK**.

Note	An error message displays if your project type does not support permanent magnets.

11. For transient and 3D eddy current projects, you may also be able to set the core loss from the **Calculate Properties for** pull-down list.
12. Click **Validate Material**.
13. Click **OK**.

    The new material is added to the local material library.

Related Topics

Defining Variable Material Properties

Importing and Converting Materials

Working with Material Libraries
Filtering Materials
Copying Materials
Removing Materials
Searching for Materials

Assigning Material Property Types

Each material property can be assigned one of the following material property types:

Simple	The material is homogeneous and linear.
Anisotropic	The material's characteristics vary with direction. • If the material property is **Anisotropic**, its characteristics are defined by its anisotropy tensor. You must define three diagonals for: anisotropic permittivity, electric loss tangent, conductivity, permeability, and magnetic loss tangent. Each diagonal represents a tensor of your model along an axis. These tensors are relative to the coordinate system specified as the object's Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently.
Nonlinear	For Relative Permittivity (valid only for 3D Electrostatic designs): • The material's relative permittivity varies according to its associated D-E curve. Refer to *Relative Permittivity for a Maxwell Material* for details on using nonlinear relative permittivity. For Bulk Conductivity (valid only for 3D DC Conduction designs): • The material's bulk conductivity varies according to its associated J-E curve. Refer to *Bulk Conductivity for a Maxwell Material* for details on using nonlinear bulk conductivity. For Relative Permeability: • The material's relative permeability varies according to its associated BH curve. Refer to *Relative Permeability for a Maxwell or RMxprt Material* for details on using nonlinear relative permeability.
Vector	For Magnetic Coercivity: • If the Relative Permeability Type is either Simple or Anisotropic, enter a Value for the Magnitude. • If the Relative Permeability Type is Nonlinear, Magnitude becomes uneditable, and three additional fields of Type Unit Vector: X Component, Y Component, and Z Component appear in which you can enter values or specify functions. For Electric Coercivity, enter a Value for the Magnitude.
Custom	Magnetostriction can be Linear, Nonlinear, or Anisotropic. Inverse

	Magnetostriction can be either Linear or Nonlinear. Refer to Magnetostriction and Inverse Magnetostriction Materials for details on using these properties.

Select a material property type from the **Type** pull-down list. Of the possibilities, only those applicable to the named material will be listed. Some material properties only use the **Simple** type. Others include several types.

- If the material property is **Anisotropic**, its characteristics are defined by its anisotropy tensor. You must define three diagonals for: relative permittivity, dielectric loss tangent, bulk conductivity, relative permeability, and magnetic loss tangent. Each diagonal represents a tensor of your model along an axis.

  These tensors are relative to the coordinate system specified as the object's Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently.
- The Relative Permeability, Relative Permittivity, and Bulk Conductivity properties can also be assigned a **Nonlinear** property type.
- Magnetostriction and Inverse magnetostriction are **Custom** properties. Refer to Magnetostriction and Inverse Magnetostriction Materials for details on using these properties.

Related Topics

Setting Coordinate Systems
Creating a Relative Coordinate System
Change the Orientation of an object
Defining Anisotropic Relative Permeability Tensors
Defining Anisotropic Relative Permittivity Tensors
Defining Anisotropic Conductivity Tensors
Defining Anisotropic Dielectric Loss Tangent Tensors
Defining Magnetic Loss Tangent Tensors
Specifying BH Curves for Nonlinear Relative Permeability
Specifying a D-E Curve for Nonlinear Relative Permittivity
Specifying a J-E Curve for Nonlinear Bulk Conductivity

Relative Permeability for a Maxwell or RMxprt Material

Select one the following for relative permeability and specify the units:

Type	Value
Simple	Type a value for the **Relative Permeability**.
Anisotropic	The following parameters appear: • T(1,1)

	• T(2,2) • T(3,3) Select either a **Simple** or **Nonlinear** Type for each of these parameters.
Nonlinear	Click **BH Curve**, and specify the coordinates for the BH-curve (or curves if multiple temperature dependencies are used) in the BH Curve dialog box. Also enables X, Y, and Z Component unit vector fields for Magnetic Coercivity.

Note	The Anisotropic type is not used in RMxprt design.

Defining Anisotropic Relative Permeability Tensors

Eddy or frequency domain solutions only.

The relative permeability of materials is assumed to be a complex quantity expressed as the following:

$$\mu = \mu' - j \cdot \mu''$$

where μ' is the real part of the complex permeability (same as the usual relative permeability), μ'' is the imaginary part of the complex permeability, and j is the imaginary unit.

The relationship above can also be written as the following:

$$\mu = \mu'\left(1 - j \cdot \frac{\mu''}{\mu'}\right) = \mu'(1 - j \cdot \tan\delta)$$

where $\tan\delta$ is the magnetic loss tangent.

For anisotropic magnetic materials, a tensor material property needs to be used. The anisotropic relative permeability tensor has the following expression:

$$\bar{\mu} = \begin{vmatrix} \mu_1'(1 - j \cdot \tan\delta_1) & 0 & 0 \\ 0 & \mu_2'(1 - j \cdot \tan\delta_2) & 0 \\ 0 & 0 & \mu_3'(1 - j \cdot \tan\delta_3) \end{vmatrix}$$

Magnetostatic, Eddy, and transient solutions only.

Maxwell allows the anisotropic magnetic material to exhibit nonlinear characteristics (magnetostatic, Eddy, and transient solutions only) on all three principle axes of the chosen coordinate system, in which case the respective nonlinear characteristics must be specified.

For example, for laminations with anisotropic behavior in the plane of the lamination, the **Anisotropic** type of material should be specified for the **Relative Permeability** property, and the two respective B-H curves should be entered. For the third direction (lamination stacking direction), one of the specified two B-H curves should also be applicable. The composition field should be set to lamination, and the stacking factor and stacking direction should be specified (see also lamination model).

General

If the material property is anisotropic, its characteristics are defined by its anisotropy tensor. You must define three diagonals for anisotropic permeability. Each diagonal represents a tensor of your model along an axis.

These tensors are relative to the coordinate system specified as the object's Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently.

1. In the **Relative Permeability** row in the **View/Edit Material** window, select **Anisotropic** from the **Type** pull-down list.

   Three rows named **T(1,1)**, **T(2,2)** and **T(3,3)** are added below the **Relative Permeability** row.
2. Enter the relative permeability along one axis of the material's permeability tensor in the **Value** box of the **T(1,1)** row.
3. Enter the relative permeability along the second axis in the **Value** box of the **T(2,2)** row.
4. Enter the relative permeability along the third axis in the **Value** box of the **T(3,3)** row.

If the relative permeability is the same in all directions, use the same values for each axis.

These values can also be defined as variables.

Related Topics

Setting Coordinate Systems

Creating a Relative Coordinate System
Change the Orientation of an object
Defining Anisotropic Relative Permittivity Tensors
Defining Anisotropic Conductivity Tensors
Defining Anisotropic Dielectric Loss Tangent Tensors
Defining Magnetic Loss Tangent Tensors

Dielectric Loss Tangent for a Maxwell Material

Specify the following for dielectric loss tangent.

Type	Value
Simple	Type a value for the**Dielectric Loss tangent**.
Anisotropic	The following parameters appear: • T(1,1) • T(2,2) • T(3,3) Type a simple value for each.

Defining Anisotropic Dielectric Loss Tangent Tensors

To represent a dielectric that dissipates the power of a high-frequency electric field, enter a dielectric loss tangent, $\varepsilon''/\varepsilon'$, property value for the material. The smaller the loss tangent, the less lossy the material.

If the material property is anisotropic, its characteristics are defined by its anisotropy tensor. You must define three diagonals for electric loss tangent. Each diagonal represents a tensor of your model along an axis.

These tensors are relative to the coordinate system specified as the object's Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently.

If electric loss tangent is anisotropic, do the following:

1. In the **Dielectric Loss Tangent** row in the **View/Edit Material** window, select **Anisotropic** from the **Type** pull-down list.

   Three rows named **T(1,1)**, **T(2,2)** and **T(3,3)** are added below the **Dielectric Loss Tangent** row.
2. Enter the ratio of the imaginary relative permittivity to the real relative permittivity in one direction in the **Value** box of the **T(1,1)** row.
3. Enter the ratio of the imaginary relative permittivity to the real relative permittivity in the second direction in the **Value** box of the **T(2,2)** row.

4. Enter the ratio of the imaginary relative permittivity to the real relative permittivity in the third orthogonal direction in the **Value** box of the **T(3,3)** row.

If the electric loss tangent is the same in all directions, use the same values for each direction.

These values can also be defined as variables.

The dielectric loss tangent may vary with frequency. To simulate the variances, define a function for the dielectric loss tangent.

Related Topics

Setting Coordinate Systems
Creating a Relative Coordinate System
Change the Orientation of an object
Defining Anisotropic Relative Permeability Tensors
Defining Anisotropic Relative Permittivity Tensors
Defining Anisotropic Conductivity Tensors
Defining Magnetic Loss Tangent Tensors

Relative Permittivity for a Maxwell Material

Specify the following for Relative Permittivity and specify the units.

Type	Value
Simple	Type a value for the **Relative Permittivity**.
Anisotropic	The following parameters appear: • T(1,1) • T(2,2) • T(3.3) Type a simple value for each.
Nonlinear	The material's relative permittivity varies according to its associated D-E curve. Refer to *Specifying a D-E Curve for Nonlinear Relative Permittivity* for details.

Related Topics

Defining Anisotropic Relative Permittivity Tensors
Specifying a D-E Curve for Nonlinear Relative Permittivity

Defining Anisotropic Relative Permittivity Tensors

Eddy or frequency domain solutions only.

The relative dielectric permittivity of dielectrics is assumed to be a complex quantity, expressed as the following:

$$\varepsilon = \varepsilon' - j \cdot \varepsilon''$$

where ε' is the real part of the complex permittivity (the same as the usual relative dielectric constant), ε'' is the imaginary part of the complex permittivity, and j is the imaginary unit.
The relationship above can also be written as the following:

$$\varepsilon = \varepsilon'\left(1 - j \cdot \frac{\varepsilon''}{\varepsilon'}\right) = \varepsilon'(1 - j \cdot \tan\delta)$$

where $\tan\delta$ is the dielectric loss tangent.
For anisotropic dielectrics, a tensor material property needs to be used. The anisotropic relative permittivity tensor has the following expression:

$$\bar{\varepsilon} = \begin{vmatrix} \varepsilon_1'(1 - j \cdot \tan\delta_1) & 0 & 0 \\ 0 & \varepsilon_2'(1 - j \cdot \tan\delta_2) & 0 \\ 0 & 0 & \varepsilon_3'(1 - j \cdot \tan\delta_3) \end{vmatrix}$$

If the material property is anisotropic, its characteristics are defined by its anisotropy tensor. You must define three diagonals for anisotropic permittivity. Each diagonal represents a tensor of your model along an axis.
These tensors are relative to the coordinate system specified as the object's Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently.

1. In the **Relative Permittivity** row in the **View/Edit Material** window, select **Anisotropic** from the **Type** pull-down list.
   Three rows named **T(1,1)**, **T(2,2)** and **T(3,3)** are added below the **Relative Permittivity** row.
2. Enter the material's relative permittivity along one tensor axis in the **Value** box of the **T(1,1)** row.

3. Enter the relative permittivity along the second axis in the **Value** box of the **T(2,2)** row.
4. Enter the relative permittivity along the third axis in the **Value** box of the **T(3,3)** row.

If the relative permittivity is the same in all directions, use the same values for each axis.

These values can also be defined as variables.

Related Topics

Setting Coordinate Systems
Creating a Relative Coordinate System
Change the Orientation of an object
Relative Permittivity for a Maxwell Material
Defining Anisotropic Relative Permeability Tensors
Defining Anisotropic Conductivity Tensors
Defining Anisotropic Dielectric Loss Tangent Tensors
Defining Magnetic Loss Tangent Tensors

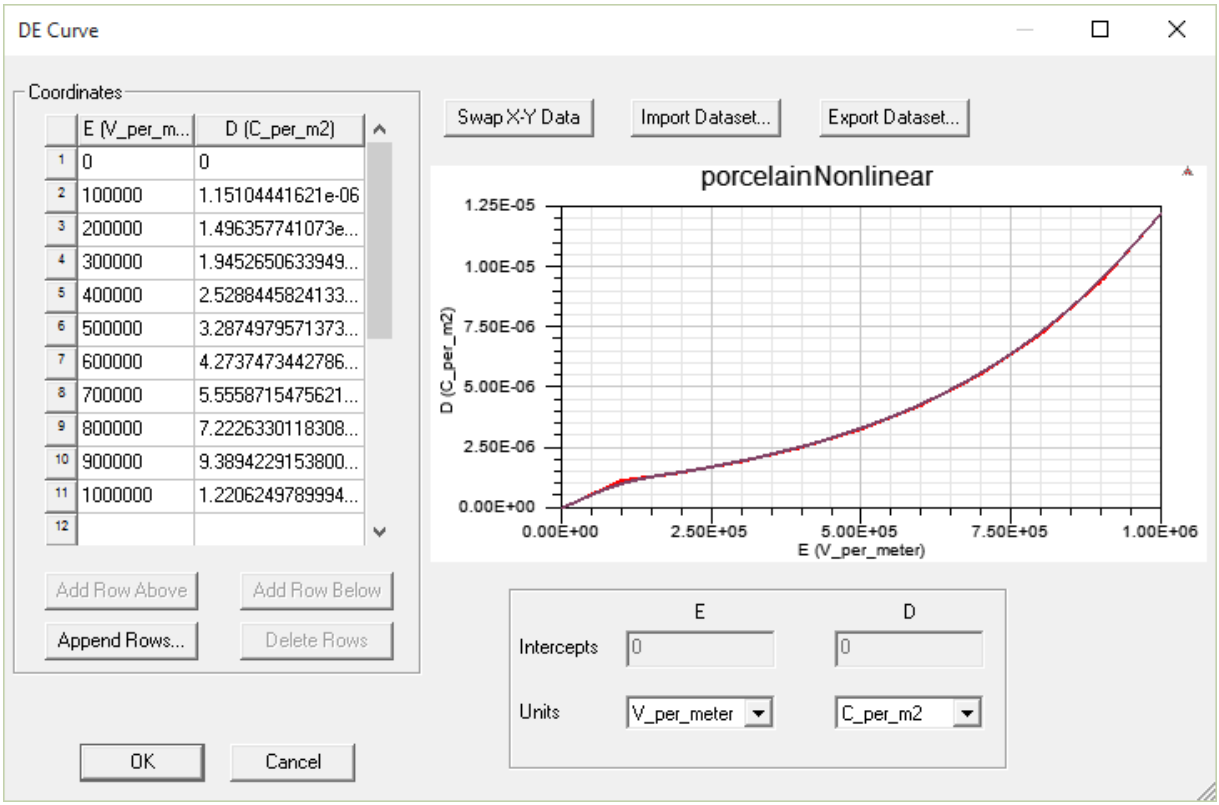
Specifying a D-E Curve for Nonlinear Relative Permittivity

Note	The D-E curve is supported only for the Maxwell 3D Electrostatic design types. A material property defined using a D-E curve will fail validation check in the other product/design types.

When you define a new material or edit an existing material in the **View /Edit Material**s window with a nonlinear relative permittivity, you need to specify the electric flux density vs. electric field strength characteristics (**D-E Curve**).

1. Open the **View / Edit Materials** dialog from the **Edit Materials** window either by:
   - Selecting an existing material that you need to edit, and click **View / Edit Material**.
   - Clicking **Add Material.**

   Any of these actions open the **View/ Edit Materials** dialog box.
2. For the **Relative Permittivity** select **Nonlinear** as the **Type**. A **D-E Curve** button appears in the **Value** column.
3. Input the D-E curve by clicking the **D-E Curve** button opens the **DE Curve** dialog box.



4. Set the **Units** for **D** and **E** by selecting from the drop down menus.

   For a material property without an existing D-E curve definition, the dialog opens with an empty list of coordinates.

5. Enter D and E values in each row of the **Coordinates** table. Placing the cursor in a table cell enables the **Add Row Above**, **Add Row Below**, and **Delete Rows** buttons.

   Note the following requirements for creating a valid curve:

   - The slope of the curve can not be less than 0.
   - The values of D and E must increase along the curve.
   - The initial values of D and E must be 0 (zero) - the curve must begin at the origin.
   - Since D-E operating points in the FEA solution may extend beyond the input D-E data set, the D-E data set is extrapolated. The slope of the last two user-defined data points is used to extrapolate the D-E curve.
   - The data points representing the D-E curve should have enough points for accurate representation of the curve. A minimum of three data points must be specified. Twenty (20) or more points should be specified with increased representation on the "knee" of the curve.

   As you enter values, the graph is updated.

   To Add or Edit rows, you can click the following buttons:

   - **Add Row Above**
   - **Add Row Below**

- **Append Rows** (to specify the number of rows to append to the table)
- **Delete Rows**

6. Optionally, click **Import Dataset** to import D-E curve data from a file, and if they are in the wrong columns, click **Swap X-Y Data** to switch the J values and E values in the graphics display. You can also use the SheetScan tool to extract curve data from sources such as manufacturer datasheets to a dataset, which can then be exported to a tab-delimited file, and imported via **Import Dataset**. (Refer to *Adding Datasets* and *Exporting Datasets* for related information on working with datasets. Refer to *Using SheetScan* for working with the SheetScan tool.)
7. When finished entering data, click **OK** to close the window.

   When you **OK** the dialog box, an error message displays if a slope is out of tolerance, identifying the out of tolerance parameter. Out of tolerance data points must be corrected before you can successfully exit the dialog.

Related Topics

Assigning Material Property Types
Relative Permittivity for a Maxwell Material

Bulk Conductivity for a Maxwell Material

Specify the following for bulk conductivity and specify the units:

Type	Value
Simple	Type a value for the **Bulk Conductivity**.
Anisotropic	The following parameters appear: • T(1,1) • T(2,2) • T(3,3) You can enter a simple value for each of these parameters.
Nonlinear	The material's bulk conductivity varies according to its associated J-E curve. Refer to *Specifying a J-E Curve for Nonlinear Bulk Conductivity* for details.

Related Topics

Defining Anisotropic Conductivity Tensors
Specifying a J-E Curve for Nonlinear Bulk Conductivity

Defining Anisotropic Conductivity Tensors

Electric conductivity can be simple or anisotropic. In either case, the corresponding values are entered in S/m (Siemens per meter).

If the material property is anisotropic, its characteristics are defined by its anisotropy tensor. You must define three diagonals for anisotropic conductivity. Each diagonal represents a tensor of your model along an axis.

These tensors are relative to the coordinate system specified as the object's Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently.

1. In the **Bulk Conductivity** row in the **View/Edit Material** window, select **Anisotropic** from the **Type** pull-down list.

   Three rows named **T(1,1)**, **T(2,2)** and **T(3,3)** are added below the **Bulk Conductivity** row.
2. Enter the conductivity along one axis of the material's conductivity tensor in the **Value** box of the **T(1,1)** row.
3. Enter the conductivity along the second axis in the **Value** box of the **T(2,2)** row.
4. Enter the conductivity along the third axis in the **Value** box of the **T(3,3)** row.

The values of the conductivity along the first and second axis apply to all axes that lie in the xy cross-section being modeled. The values of the conductivity along the third axis applies to the z-component. These values affect current flowing in dielectrics between the conductors.

These values can also be defined as variables.

Related Topics

Setting Coordinate Systems

Creating a Relative Coordinate System

Change the Orientation of an object

Bulk Conductivity for a Maxwell Material

Defining Anisotropic Relative Permeability Tensors

Defining Anisotropic Relative Permittivity Tensors

Defining Anisotropic Dielectric Loss Tangent Tensors

Defining Magnetic Loss Tangent Tensors

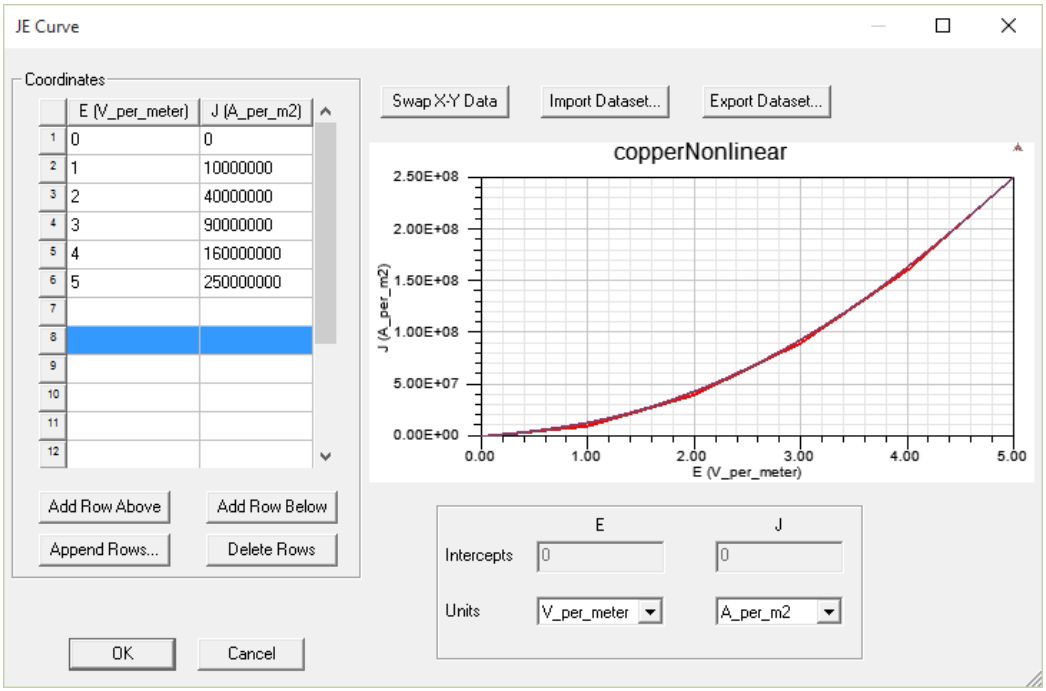
Specifying a J-E Curve for Nonlinear Bulk Conductivity

Note	The J-E curve is supported only for the Maxwell 3D DC Conduction design types. A material property defined using a J-E curve will fail validation check in all the other product/design types.

When you define a new material or edit an existing material in the **View /Edit Material**s window with a nonlinear bulk conductivity, you need to specify the current density vs. electric field characteristics (**J-E Curve**).

1. Open the **View / Edit Materials** dialog from the **Edit Materials** window either by:
   - Selecting an existing material that you need to edit, and click **View / Edit Material**.
   - Clicking **Add Material.**

   Any of these actions open the **View/ Edit Materials** dialog box.
2. For the **Bulk Conductivity** select **Nonlinear** as the **Type**. A **J-E Curve** button appears in the **Value** column.
3. Input the J-E curve by clicking the **J-E Curve** button opens the **JE Curve** dialog box.



4. Set the **Units** for **J** and **E** by selecting from the drop down menus.

   For a material property without an existing J-E curve definition, the dialog opens with an empty list of coordinates.
5. Enter J and E values in each row of the **Coordinates** table. Placing the cursor in a table cell enables the **Add Row Above**, **Add Row Below**, and **Delete Rows** buttons.

   Note the following requirements for creating a valid curve:
   - The slope of the curve can not be less than 0.
   - The values of J and E must increase along the curve.
   - The initial values of J and E must be 0 (zero) - the curve must begin at the origin.
   - Since J-E operating points in the FEA solution may extend beyond the input J-E data set, the J-E data set is extrapolated. The slope of the last two user-defined data points is used to extrapolate the J-E curve.
   - The data points representing the J-E curve should have enough points for accurate representation of the curve. A minimum of three data points must be specified. Twenty

(20) or more points should be specified with increased representation on the "knee" of the curve.

As you enter values, the graph is updated.

To Add or Edit rows, you can click the following buttons:

- **Add Row Above**
- **Add Row Below**
- **Append Rows** (to specify the number of rows to append to the table)
- **Delete Rows**

6. Optionally, click **Import Dataset** to import J-E curve data from a file, and if they are in the wrong columns, click **Swap X-Y Data** to switch the J values and E values in the graphics display. You can also use the SheetScan tool to extract curve data from sources such as manufacturer datasheets to a dataset, which can then be exported to a tab-delimited file, and imported via **Import Dataset**. (Refer to *Adding Datasets* and *Exporting Datasets* for related information on working with datasets. Refer to *Using SheetScan* for working with the SheetScan tool.)
7. When finished entering data, click **OK** to close the window.

   When you **OK** the dialog box, an error message displays if a slope is out of tolerance, identifying the out of tolerance parameter. Out of tolerance data points must be corrected before you can successfully exit the dialog.

Related Topics

Assigning Material Property Types

Bulk Conductivity for a Maxwell Material

Magnetic Loss Tangent

To represent a magnetic material such as ferrite that dissipates the power of a high-frequency magnetic field, enter a magnetic loss tangent, μ''/μ', property value for the material. The smaller the loss tangent, the less lossy the material.

Note	If you plan to do a fast frequency sweep for a design that includes dielectrics, make sure that the dielectric or magnetic loss tangent does not vary significantly over the requested frequency range. If they do, the results may not be what you expect. In cases where the loss tangent does vary significantly over the frequency range in which you are interested, copy and solve the design several times, adjusting the loss tangent and associated frequency range for the copied design so that the loss tangent is relatively stable over the design's requested frequency range.

Note	When magnetic loss tangent is used, the resulting hysteresis core loss will be reflected in the resistance term for both the impedance matrix and the Simplorer Dynamic Eddy Current ROM.

Imaginary Permeability

Eddy Current

Some materials exhibit a permeability that includes both a real and imaginary component. The imaginary component is used to model magnetic losses in a time-varying field using the relationship:

$$B = (\mu_r' - j(\mu_r''))\mu_0 H$$

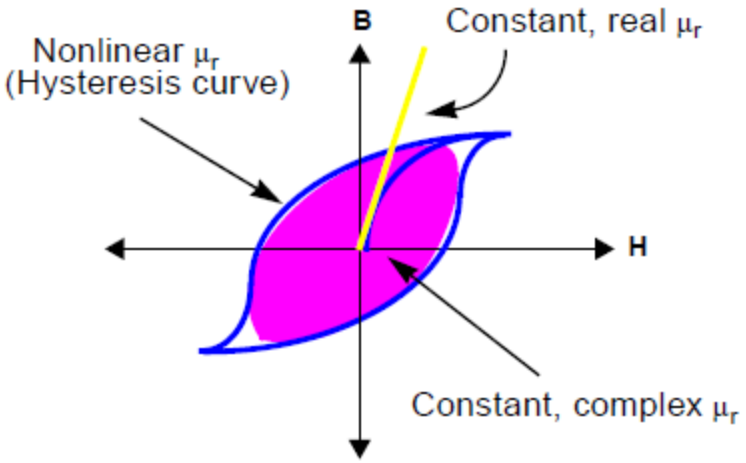
where:

μ_r' is the real component of the relative permeability.

μ_r'' is the imaginary component of the relative permeability.

μ_0 is the permeability of free space.

As shown below, a complex relative permeability causes the B field to lag behind the H field – similar to the behavior of a nonlinear, lossy material. The power loss during this cycle (the color shaded area) is approximately equal to the hysteresis loss (the area within the blue lines). The hysteresis curve for a material with a constant, real permeability (the straight yellow line) is shown as a reference.



When the nature of losses is also due to magnetic hysteresis, Maxwell allows you to model such effects using complex permeability by allowing the specification of the tangent of magnetic loss. When a linear material exhibits hysteresis effects, it should be characterized using a complex permeability, as below:

$$\underline{\mu} = \mu' - j\mu'' = \mu'\left(1 - j\tan\delta\right)$$

where $\tan\delta$ is the magnetic loss tangent, $\frac{\mu''}{\mu'}$

Enter the imaginary relative permeability of a material, μ'', in the **Imag. Permeability** field. The default imaginary permeability of zero is that of a material that exhibits no magnetic loss in a time-varying field.

Defining Magnetic Loss Tangent Tensors

If the material property is anisotropic, its characteristics are defined by its anisotropy tensor. You must define three diagonals for anisotropic magnetic loss tangent. Each diagonal represents a tensor of your model along an axis.

These tensors are relative to the coordinate system specified as the object's Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently.

1. In the **Magnetic Loss Tangent** row in the **View/Edit Material** window, select **Anisotropic** from the **Type** pull-down list.

   Three rows named **T(1,1)**, **T(2,2)** and **T(3,3)** are added below the **Magnetic Loss Tangent** row.
2. Enter the ratio of the imaginary relative permeability to the real relative permeability in one direction in the **Value** box of the **T(1,1)** row.
3. Enter the ratio of the imaginary relative permeability to the real relative permeability in the second direction in the **Value** box of the **T(2,2)** row.
4. Enter the ratio of the imaginary relative permeability to the real relative permeability in the third direction in the **Value** box of the **T(3,3)** row.

If the magnetic loss tangent is the same in all directions, use the same values for each direction. These values can also be defined as variables.

Related Topics

Setting Coordinate Systems

Creating a Relative Coordinate System

Change the Orientation of an object

Defining Anisotropic Relative Permeability Tensors

Defining Anisotropic Relative Permittivity Tensors

Defining Anisotropic Conductivity Tensors

Defining Anisotropic Dielectric Loss Tangent Tensors

Defining Variable Material Properties

When defining or modifying a material's properties, each material property value in the **View/Edit Material** window can be assigned a project variable. Simply type the project variable's name in the appropriate **Value** box. Project variables are used for material properties because materials are stored at the project level.

For example, define a project variable with the name **MyPermittivity** and define its value as **4**. To assign this property value to a material, type **$MyPermittivity** in the **Value** box for the material. Be sure to include the prefix **$** before the project variable name, which notifies Maxwell that the variable is a project variable.

Note	By default, not all of the available properties are displayed in the materials table. Only the properties commonly used by the product are displayed. To view the complete table of properties, see Filtering Materials.

Related Topics

Working with Variables

Assigning Material Property Types

Specifying Thermal Modifiers

Specifying Thermal Modifiers

To specify Thermal modifiers for a material:

1. In the View/ Edit materials dialog you must enable the View/ Edit Modifier check box for Thermal Modifier.

   This causes the Properties of the Material table to expand to include a Thermal Modifier column. By default, the Thermal Modifier property is set to **None**.
2. Selecting **Edit...** from the drop down menu, rather than **None** causes the **Edit Thermal Modifier** dialog to appear.
3. Select **Expression** radio button to display the **Parameters Modifier** text field or the **Quadratic** radio button to display the tabs for **Basic Coefficient Set** and **Advanced Coefficient Set**.
   - With **Expression** selected, you can write an equation for a thermal modifier in the Parameters **Modifier** text field.

     Checking **Use temperature dependent data set** disables the Modifier text field. You can then use the drop down menu to select Add/Import Dataset. This lets you define the thermal modifier as a data set.
   - With the **Quadratic** radio button selected, in the **Basic Coefficient** tab, you can edit fields for the TempRef and units, and fields for C1 and C2 for the following equation:

     P(Temp) = Pref[1+ C1(Temp - TempRef) + C2(Temp - TempRef)^2]

     where the Pref is defined as the reference relative permittivity.
   - With the **Quadratic** radio button selected, in the **Advanced Coefficient Set** tab, you can edit fields for lower and upper temperature limits (TL and TU respectively) and select their units from the drop down.

     You can also edit the constant value limit for the thermal modifier values outside the limits. By default, these are automatically calculated. Uncheck the Auto Calculate TML and TMU to specify new values for thermal modifier lower (TML) and thermal modifier upper (TMU).

4. Click **OK** to accept the edits and return to the View/ Edit materials dialog.

Related Topics

Adding Datasets

View/ Edit materials dialog

Irreversible Demagnetization Due to Temperature Change

Defining Material Properties as Expressions

When defining or modifying a material's properties, each material property value in the **View/Edit Material** window can be assigned a mathematical expression. Simply type the expression in the appropriate **Value** box. Expressions typically contain intrinsic functions, such as sin(x), and arithmetic operators, such as +, -, *, and /, but do not include project variables.

Note	By default, not all of the available properties are displayed in the materials table. Only the properties commonly used by the product are displayed. To view the complete table of properties, see Filtering Materials.

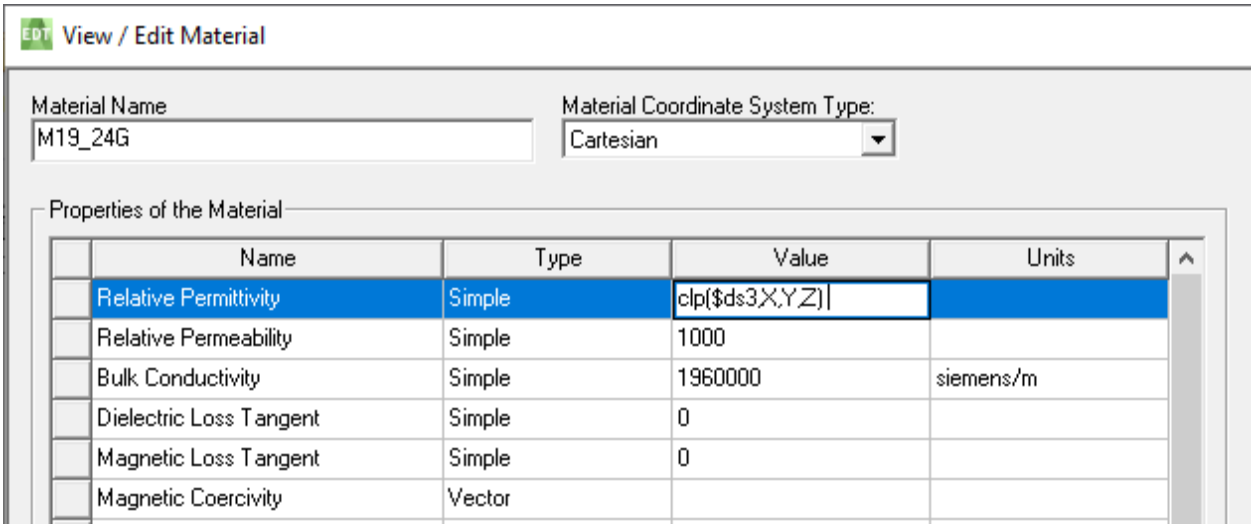
Related Topics

Defining Variable Material Properties

Defining Functional Material Properties

Any material property that can be specified by entering a constant can also be specified using a mathematical function. This is useful when you are defining a material property whose value is given by a mathematical relationship — for instance, one relating it to frequency or another property's value. When defining or modifying a material's properties, simply type the name of the function in the appropriate **Value** box.

Optionally, for 2D and 3D transient designs, you can enter an expression for a 3D(X,Y,Z) dataset in the Value box for the properties of conductivity, permeability, and permittivity, as shown in the figure below. The 3D dataset models the spatially varying material property change with respect to location along the object. For detailed information on using 3D(X,Y,Z) datasets, please refer to *Setting a Spatial Modifier*.



Note	By default, not all of the available properties are displayed in the materials table. Only the properties commonly used by the product are displayed. To view the complete table of properties, see Filtering Materials.

Related Topics

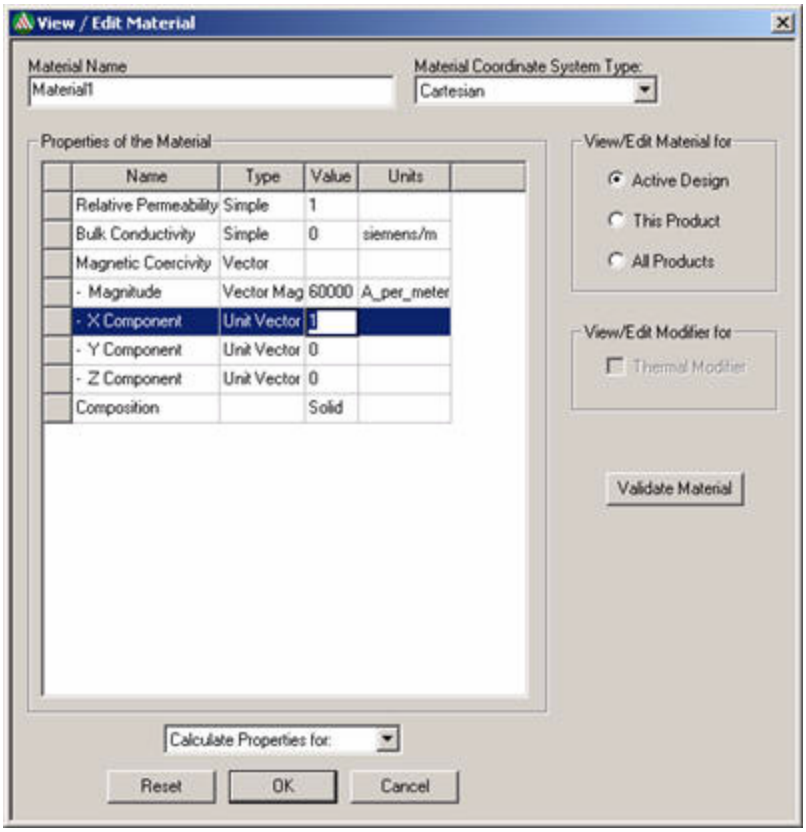
Defining Mathematical Functions
Defining Variable Material Properties
Setting a Spatial Modifier
Adding 3D Datasets

Material Coordinate Systems

Maxwell supports the following three types of coordinate systems, which can be used to define some vector or anisotropic material properties:

- Cartesian (defined by the X, Y, and Z axes)
- Cylindrical (defined by the R, Phi, and Z axes)

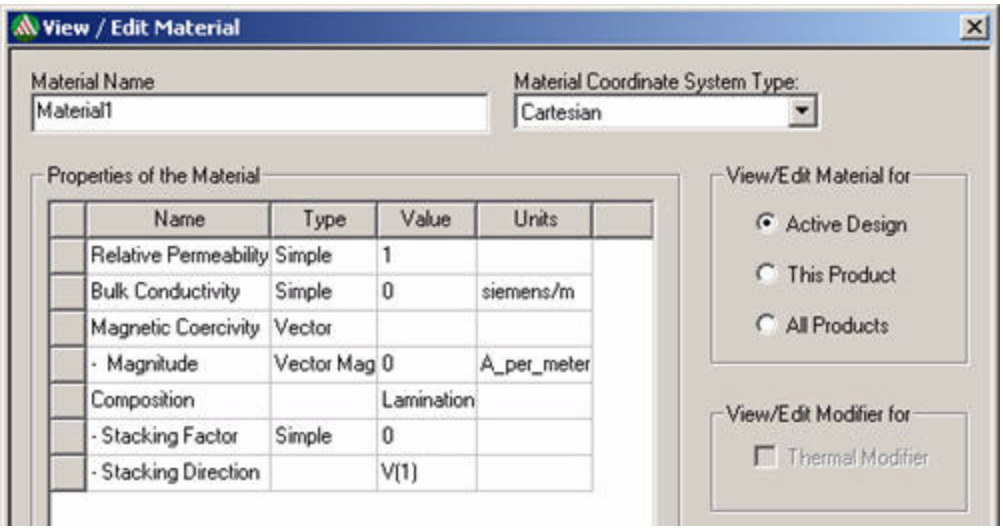
- Spherical (defined by the Rho, Theta, and Phi axes)



To get the fields for the vector magnetic coercivity vector, a non-zero value must first be specified for the magnitude of the vector. The magnitude can also be functional. The magnitude of the magnetic coercivity is expected to be < 0. If you use a positive value, the direction used will be the opposite of the unit vector specified.

Lamination Modeling

Lamination modeling allows you to specify a stacking factor and stacking direction, which represents the direction perpendicular to the plane of the lamination. Cartesian, cylindrical, or spherical coordinate systems can be used to specify the stacking direction.



This lamination model is an alternative way of specifying anisotropic behavior when using laminations (which is a special case that is frequently done). For the frequently encountered case of isotropic laminations (where the global model is anisotropic due to the existence of laminations but the laminations themselves are isotropic), the above picture shows a possible setup. Choose a nonlinear behavior for the material of the lamination with a user specified B-H curve, while the global anisotropy is modeled by specifying a stacking factor and the stacking direction, which is perpendicular to the plane of laminations. In this way, Maxwell can consider a global anisotropy with two orientations — one in the plane of the lamination, and the other in the corresponding orthogonal direction.

Note	If a stacking factor less than 1 is applied to a magnetic core, the flux density plotted will represent the AVERAGE flux density in both the air and the steel (and not the flux density in the steel laminations). The average flux density is reduced as the stacking factor is reduced.
Note	Lamination modeling is supported for magnetostatic, transient solutions and eddy current, where the stacking value is between 0 and 1. Please use appropriate materials with suitable lamination definitions for the solver. Other solvers will ignore the lamination definition in the material.
Note	When a stacking factor (SF) is used, users need to modify the electrical steel core loss coefficients based on the stacking factor as shown below. As the stacking factor decreases < 1, the core loss coefficients increase. Kh_scaled = Kh / SF Kc_scaled = Kc / SF Ke_scaled = Ke / sqrt(SF)

Related Topics

Stacking Direction

Core Loss Model for a Maxwell Material

Stacking Direction

When defining a lamination model, the stacking direction must be specified. The direction depend upon the Material Coordinate System Type chosen as follows:

	Cartesian Coordinates	Cylindrical Coordinates	Spherical Coordinates
V (1)	X	r	r
V (2)	Y	θ	θ
V (3)	Z	Z	ϕ

Litz Wire Modeling

Note	Litz wire modeling is currently supported for windings only (not for Conductor current sources) in Maxwell 2D/3D Transient, and 2D/3D Eddy Current designs.

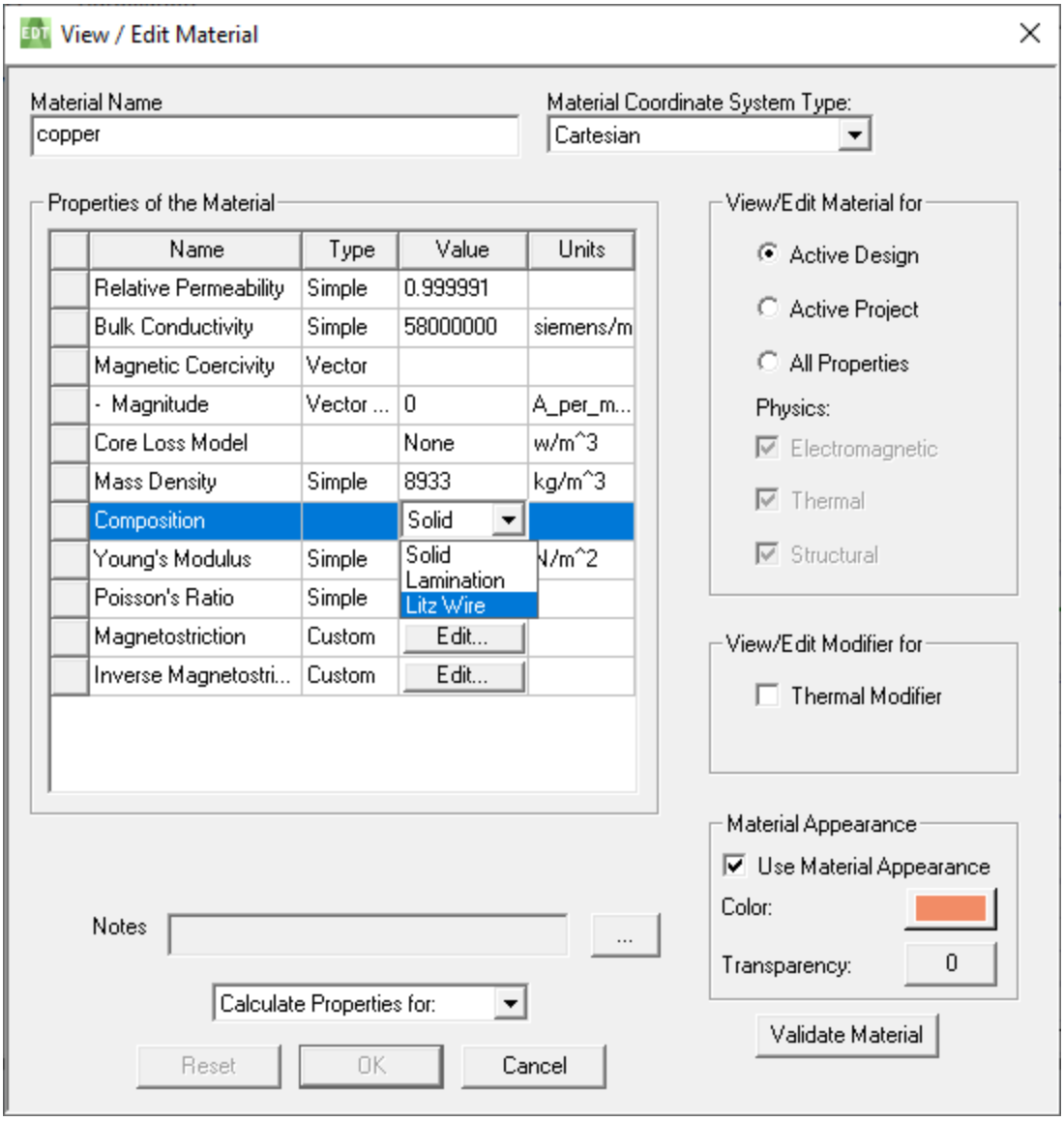
Litz wire is a multi-strand wire or cable whose primary benefit is in reducing AC losses in high frequency windings. It consists of many thin, individually insulated, wire strands, which are twisted or woven together in carefully prescribed patterns. The multiple strand construction of Litz wire allows current to divide uniformly between strands, thus reducing skin effect and proximity effect losses.

Some basic assumptions for applying this modeling approach are:

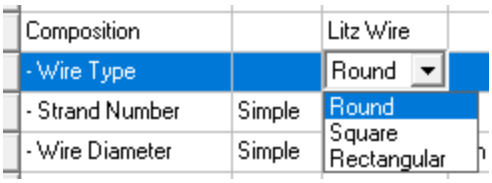
- Each Litz wire strand is insulated individually.
- Each strand is continuously and perfectly transposed so that every strand occupies every location in the cross-sectional area of the bundle over the lay length.
- The largest dimension of strands (e.g., diameter or thickness) is smaller than the skin depth δ for the strand's material.

$$\delta = 1 \Big/ \sqrt{\pi f \mu \sigma}$$

Lizt wire modeling allows you to specify the wire type, number of strands, and various other parameters that define the shape of the Litz wire bundle.



When Litz Wire is selected, **Wire Type** is displayed under **Composition**. There are three options **Round**, **Square**, and **Rectangular**, for pull-down menu Wire Type. These are the three possible shapes for a Litz wire.



Note	If **Litz Wire** is selected and any wire dimension is 0 , the **Composition** material property will be treated as **Solid**, and a warning message to that effect will appear in the Message

	Manager.

If **Round** (the default) is selected, **Strand Number** and **Wire Diameter** are displayed. Default values are as shown below. **Strand Number** is the integer value of the number of strands in the wire bundle. **Wire Diameter** is a real number that defines the diameter of a single wire in the wire bundle. You can also select the desired unit of measure.

Composition		Litz Wire	
- Wire Type		Round	
- Strand Number	Simple	1	
- Wire Diameter	Simple	0	mm

If **Square** is selected for **Wire Type**, **Strand Number** and **Wire Width** are displayed. Default values are as shown below. **Strand Number** is the integer value of the number of strands in the wire bundle. **Wire Width** is a real number that defines the width of a single wire in the wire bundle. You can also select the desired unit of measure.

Composition		Litz Wire	
- Wire Type		Square	
- Strand Number	Simple	1	
- Wire Width	Simple	0	mm

If **Rectangular** is selected for **Wire Type**, the properties **Strand Number**, **Wire Width**, **Wire Thickness**, **Thickness Direction** and **Width Direction** will be displayed. Default values are as shown below. For thickness direction and width direction The drop-down menus include three options: V(1), V(2) and V(3), representing x, y, and z directions, respectively, for the **Cartesian** Material Coordinate System Type; and r, θ, z directions, respectively, for the **Cylindrical** Material Coordinate System Type.

Composition		Litz Wire	
- Wire Type		Rectangular	
- Strand Number	Simple	1	
- Wire Width	Simple	0	mm
- Wire Thickness	Simple	0	mm
- Thickness Direction		V(1)	
- Width Direction		V(2)	

For inner/outer core types, the armature ac or dc windings inside core slots for large capacity machines rotating around z-axis usually have coils with wire thickness in r direction, and width in θ direction or, rarely, have coils with wire thickness in θ direction, and width in r direction.

For axial-flux core types, the armature ac or dc windings inside core slots of large capacity machines rotating in z-axis usually have coils with wire thickness in z direction, and width in θ direction, or rarely have coils with wire thickness in θ direction, and width in z direction.

The total loss from the Litz wire model (including DC loss and AC loss) is applied for thermal coupling. The effect of temperature on Litz wire loss calculation is taken into account in Maxwell by setting a thermal modifier for the bulk conductivity material property.

Core Loss Model for a Maxwell Material

Specify the following parameters and units for core loss model:

Note	All core loss property parameters for **Electrical Steel** and **Power Ferrite** materials can be temperature-dependent, as defined using a thermal modifier.

Name	Value
None	No core loss is to be calculated for this material.
Electrical Steel	The following parameters appear: • **Kh** – Type can be either **Simple** or **Anisotropic** • **Kc** – Type can be either **Simple** or **Anisotropic** • **Ke** – Type can be either **Simple** or **Anisotropic** • **Kdc** (not used for Eddy Current solution type) **Note**: In 2019.R1 and earlier versions, the default value for Kdc is 0.65, that is, if it is set as 0 in the UI, solvers will handle it as 0.65. In 2019.R2 and later versions, solvers will get the value as it is specified. • **Equiv. Cut Depth** (Refer to *Cut Edge Effects on Core Loss for Lamination Steels* for details on the use of this parameter.) For **Kh**, **Kc**, and **Ke**, if the type is anisotropic, The following parameters appear: • T(1,1) • T(2,2) • T(3,3) These characteristics are defined by their anisotropy tensors. Each diagonal represents a tensor of your model along an axis. You can enter a simple value for each of these parameters. These tensors are relative to the coordinate system specified as the object's Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently. **Note**: Selecting **Electrical Steel** also enables the Calculate Properties for Core Loss Coefficient pull-down list at the bottom of the dialog box. Selecting either **Core Loss at One Frequency** or **Core Loss versus Frequency** allows you to edit the **BP Curve**.
Power Ferrite	The following parameters appear: • **Cm** – Type can be either **Simple** or **Anisotropic** • **X** – Type can be either **Simple** or **Anisotropic**

<table>
<tr><td></td><td>

Y – Type can be either Simple or Anisotropic
Kdc (not used for Eddy Current solution type)
<table>
<tr><td>Note</td><td>In 2019.R1 and earlier versions, the default value for Kdc is 0.65, that is, if it is set as 0 in the UI, solvers will handle it as 0.65. In 2019.R2 and later versions, solvers will get the value as it is specified.</td></tr>
</table>

Equiv. Cut Depth (Refer to Cut Edge Effects on Core Loss for Lamination Steels for details on the use of this parameter.)

For Cm, X, and Y, if the type is anisotropic, The following parameters appear:

T(1,1)
T(2,2)
T(3,3)

These characteristics are defined by their anisotropy tensors. Each diagonal represents a tensor of your model along an axis. You can enter a simple value for each of these parameters.
These tensors are relative to the coordinate system specified as the object’s Orientation property. By specifying different orientations, several objects can share the same anisotropic material but be oriented differently.
<table>
<tr><td>Note</td><td>

Selecting Power Ferrite also enables the Calculate Properties for Core Loss Coefficient pull-down list at the bottom of the dialog box. Selecting either Core Loss at One Frequency or Core Loss versus Frequency allows you to edit the BP Curve.
In Transient Solver, X must be less than Y.

</td></tr>
</table>
</td></tr>
<tr><td>B-P Curve</td><td>
The following parameter appears:

Core Loss. Click the B-P Curve button to define temperature-dependent core loss curves. (Refer to Calculating Properties for Temperature-dependent Core Loss Curves in Maxwell for details on the use of this parameter.)

If view/edit parameters for Active Project is selected, the following additional parameters appear:

Kdc Coefficient Considering DC Bias Effects
Equiv. Cut Depth (Refer to Cut Edge Effects on Core Loss for Lamination Steels for details on the use of this parameter.)

</td></tr>
<tr><td>Hysteresis Model</td><td>
The following parameters appear:

Intrinsic Coercivity Hci
Remanence Br
Kc
Equiv. Cut Depth (Refer to Cut Edge Effects on Core Loss for Lamination Steels for details on the use of this parameter.)

</td></tr>
</table>

	Note	Selecting **Hysteresis Model** also enables the Calculate Properties for Hysteresis Loop pull-down list item at the bottom of the dialog box. Selecting **Hysteresis Loop** opens the **Properties for Hysteresis Loop** window.

Some additional considerations:

DC-biased effects

The core loss computation is based on the traditional three core loss coefficients **Kh**, **Kc**, and **Ke**, plus the optional **Kdc**. In order to properly consider the impact of DC-bias on core loss, an additional factor derived from Kdc is applied to scale the hysteresis loss from the Bertotti's formula. The additional factor is expressed as Kph = 1 + Kdc * Bdc^2. If the user leaves the default value of Kdc as zero, this indicates that the impact of DC-bias will not be considered because of the additional factor Kph = 1. If a user wants to consider the DC-bias effect, the user needs to do an experiment to measure core loss at various DC-bias flux densities Bdc, and extract Kdc based on the Kph expression as given above using a linear regression. If the measured data are not available, you can input 0.65 for Kdc.

Note	The use of Kdc when computing DC-biased effects has no effect on the hysteresis model-based core loss computation approach since in such a case, the hysteresis model has inherently consider the impact of DC-biased effects on hysteresis loss.

Hysteresis model-based core loss computation approach

Using this approach, the computation of the hysteresis loss component is based on the input of a hysteresis loop, and the computation of the eddy current loss component based on the classic eddy current loss coefficient Kc. This means that the impact of classic excess loss must be included in the hysteresis loss (sometime called: dynamic hysteresis loss). There are two ways to enable computation of core loss for the hysteresis model-based approach:

- Define the descending limiting branch of the hysteresis loop in the nonlinear BH curve input panel together with the input of **Kc**, which is defined by choosing **Electrical Steel** as the **Core Loss Model** on the **Properties of the Material** panel. Please note that any values entered for **Kh** and **Ke** will be ignored.

Note	This approach requires use of a material that has a vector magnitude direction set to (0,0,0) to enable the hysteresis loss calculation; and must also have its **Core Loss Setting** checked in the **Set Core Loss** dialog to include the eddy current component of core loss.

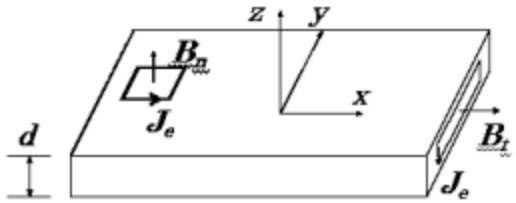
- Define the hysteresis loop by the input of a normal nonlinear BH curve together with the selection of **Hysteresis Model** as the **Core Loss Model** and the input of values for **Intrinsic Coercivity Hci**, **Remanence Br** (optional) and **Kc**.

Note	The second approach is able to more accurately model minor loop loss behavior.

Additional core loss due to flux normal to laminations

When the lamination model is used in the Maxwell 3D transient solver, an additional eddy current component of core loss will be automatically calculated due to the normal component of flux (B_n)

on the lamination stack and will be added to the total core loss, if conductivity > 0 has been specified in the material manager. The basic idea of the algorithm for considering this additional eddy current component of core loss is to force induced eddy current only occurring in the plane of lamination by introducing anisotropic conductivity.



Lamination stacked in Z direction

To compute this additional core loss component:

- The material conductivity which is taken from the project material library (not from the core loss input panel) must be specified as non-zero.
- The object must be defined as a lamination.
- The eddy effect must always be turned off for laminated objects.
- Core Loss Setting should be activated on the Set Core Loss panel, General tab under Excitations.
- A fine mesh in the object is required to accurately calculate the additional eddy current loss due to this effect.
- For adjacent lamination stacks that are touching, insulating boundaries should be applied to prevent eddy currents from circulating between the stacks.

Additional core loss due to manufacturing processes

Lamination steel is cut to the final shape by manufacturing processes such as punching or laser cutting. These processes deteriorate the magnetic properties of the material due to the plastic deformation and residual stress near the cut edge that directly cause core loss increase in the deformation region. These losses can be taken into account for the accurate prediction of the core losses using the **Equiv. Cut Depth** property value.

To compute this additional core loss component:

- The Edge Cut Based mesh operation must be specified;
- The **Composition** property for the assigned material must be set to **Lamination**;
- The **Equiv. Cut Depth** material property must be set to a value greater than zero (default is 1mm).

Related Topics

Calculating Properties for Core Loss (BP Curve)

Core Loss Coefficients for Electrical Steel

Core Loss Coefficient Extraction

Setting Core Loss for Transient and Eddy Current Solvers
Lamination Modeling
Setting a Thermal Modifier
Setting the Temperature of Objects

Cut Edge Effects on Core Loss for Lamination Steels

Lamination steel is cut to the final shape by manufacturing process such as punching and laser cutting, which deteriorates the magnetic properties of the material. This is because the plastic deformation and residual stress near the cut edge cause core loss increase in the deformation region. These losses should be taken into account for the accurate prediction of core losses.

Cut-edge loss model

Assume core loss per unit volume without considering cut-edge effects is p_0.

If the cut-edge deformation depth is d_c, the cut-edge area is S_c, and the core loss increasing factor is k_0, the core loss considering cut-edge effects will be:

$$P_c = (p_0 + k_0 p_0)\, d_c\, S_c$$

Thus, the loss increase due to cut-edge effects is:

$$\Delta P_c = p_0 (k_0 d_c) S_c$$

Since parameters d_c and k_0 cannot normally be derived individually, we introduce an equivalent cut-edge depth:

$$d_{eq} = k_0 d_c$$

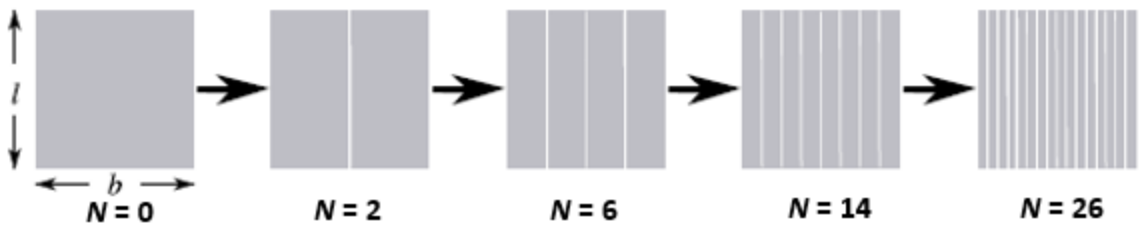
such that d_{eq} can be identified by:

$$d_{eq} = \Delta P_c / (p_0 S_c)$$

Parameter d_{eq} derivation

Based on the measurement of core loss of lamination sheets with variable width to consider the cut-edge effects, manufacturers can use single sheet testing method to measure the impact of cut-edges on core loss. The steps for deriving the parameter d_{eq} are as follows:

- Cut lamination to N edges of different widths as shown below.



- Measure the loss *Pij* with different width (*j*) under different magnetic polarization (*i*).

- Derive the loss *Pij'* using the cut-edge loss model to obtain one equation for every combination of *i* and *j* with parameter d_{eq} to be identified:

$$P_{ij}' = P_{i0} + \Delta P_{ij} = P_{i0}(1 + d_{eq} N_j / b)$$

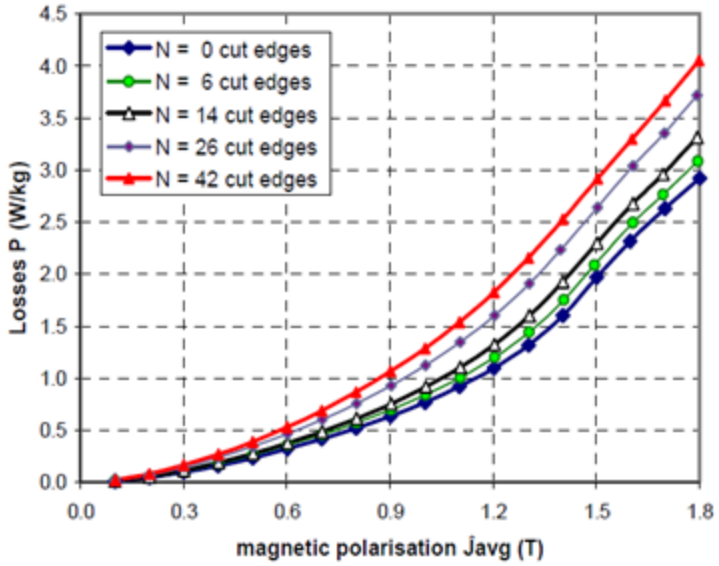
- Find the error ε between derived and measured:

$$\varepsilon = \sum_{i=1}^{n} \sum_{j=1}^{m} (P_{ij}' - P_{ij})^2$$

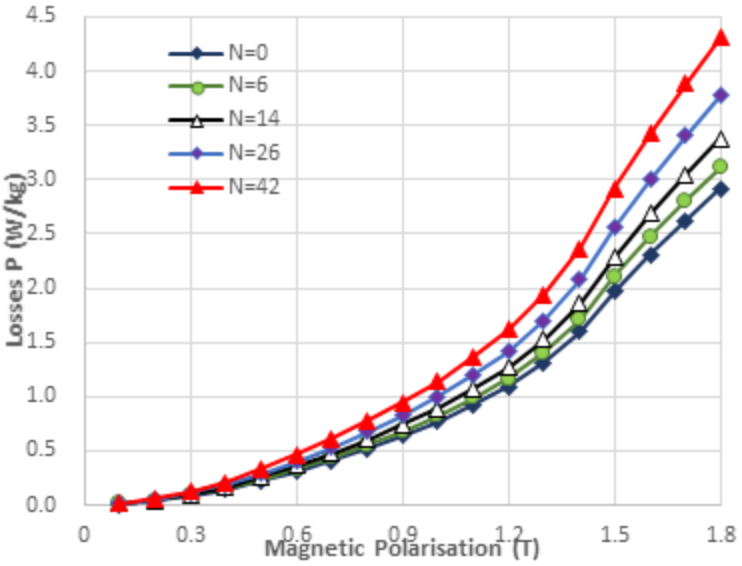
- Derive parameter d_{eq} by minimizing the error:

$$d_{eq} = b \frac{\sum_i \sum_j (P_{ij} - P_{i0}) P_{i0} N_j}{\sum_i \sum_j (P_{i0} N_j)^2}$$

For example, the following shows the *measured* core loss varying with magnetic polarization for several values of the number of cut edges;

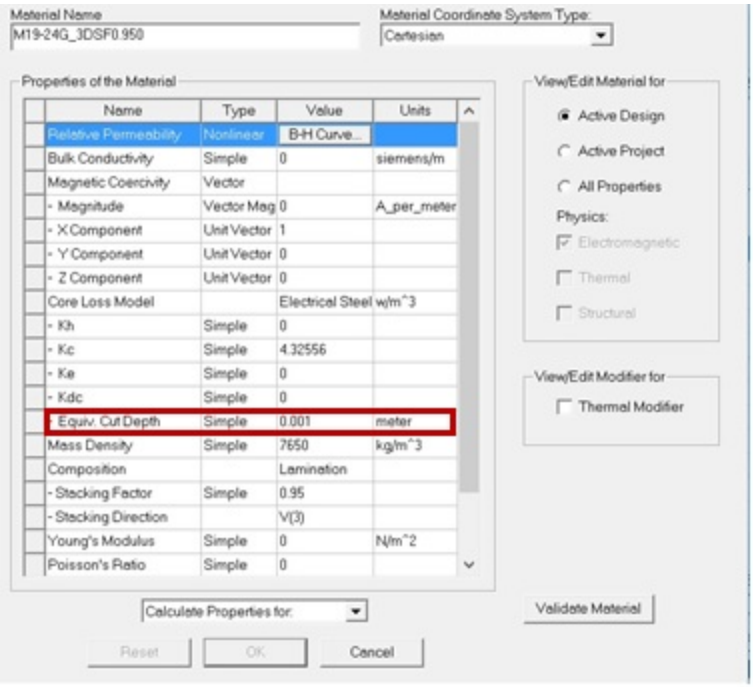


and the following shows the *simulated* core loss varying with magnetic polarization for several values of the number of cut edges



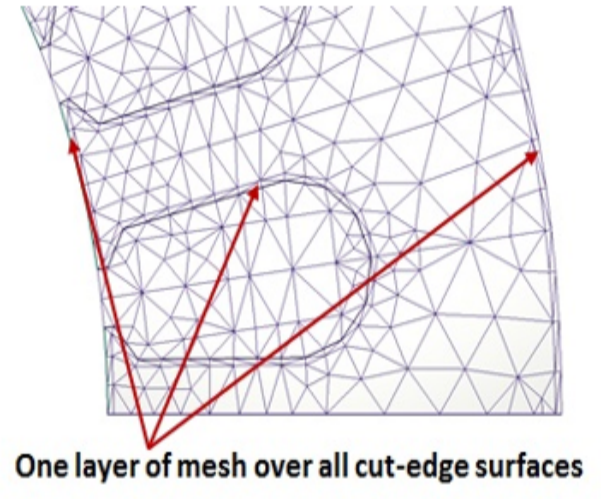
Maxwell Implementation of Cut Edge Effects

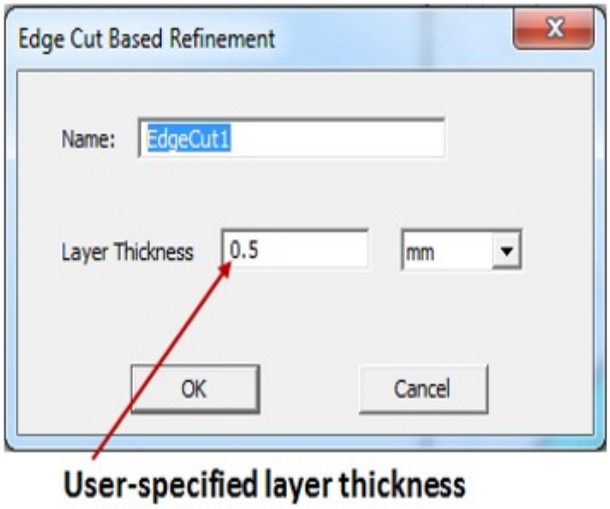
Maxwell uses the parameter d_{eq} described above as an additional input material property, **Equiv. Cut Depth**.



To implement cut edge effects, the Maxwell solver:

1. Constructs one layer of mesh over all cut-edge surfaces with user-specified layer thickness d_{LT}





2. The solver then computes core loss considering cut-edge effects P_c on all elements inside the layer:

$$\begin{aligned} P_c &= P_0 + \Delta P_c \\ &= p_0\, d_{LT}\, S_c + p_0\, d_{eq}\, S_c \\ &= \left(1 + \frac{d_{eq}}{d_{LT}}\right) p_0\, d_{LT}\, S_c \\ &= k_c\, P_0 \end{aligned}$$

Additional Considerations

It is not necessary for the layer thickness d_{LT} to be equal to the equivalent cut-edge depth d_{eq} since the above equation is able to ensure the derived core loss increase matches the measurements.

To take into account the cut-edge effects in core loss computation, the following three conditions must be fulfilled at the same time:

- The Edge Cut Based mesh operation must be specified;
- The **Composition** property for the assigned material must be set to **Lamination**;
- The **Equiv. Cut Depth** material property must be set to a value greater than zero (default is 1 mm).

Related Topics

Core Loss Model for a Maxwell Material

Calculating Properties for Core Loss (BP Curve)

To be able to extract parameters from the loss characteristics (B-P Curve), you first set the **Core Loss Model** of the material to **Electrical Steel** or **Power Ferrite** as a material property in the **View / EditMaterial** dialog box.

Note	• If the Core Loss Model is set to **Hysteresis Loop**, refer to *Calculating Properties Using a Hysteresis Loop in Maxwell*. • If the Core Loss Model is set to **B-P Curve,** refer to *Calculating Properties for Temperature-dependent Core Loss Curves in Maxwell*.

To calculate core loss properties for an electrical steel material:

1. Click **Tools>Edit Configured Libraries>Materials**.
   - Alternatively, in the project tree, you can right-click **Materials**, under **Definitions**, and select **Edit All Libraries**.

   The **Edit Libraries** dialog box appears.
2. Click **Add Material**.
   The **View / Edit Material** dialog box appears.
3. In the **Core Loss Model** row, select **Electrical Steel** or **Power Ferrite** from the **Value** pull-down list.

   Additional parameters appear in the following table **Core Loss Model** (**Kh**, **Kc**, **Ke**, **Kdc**, and **Equiv. Cut Depth** for electrical steel, and **Cm**, **X**, **Y**, **Kdc**, and **Equiv. Cut Depth** for power ferrite). For **Electrical Steel**, the **Core Loss Coefficient** selection in the **Calculate Properties for** pull-down list at the bottom of the dialog box is also enabled. This allows the electrical steel coefficients **Kh**, **Kc**, **Ke**, and **Kdc**, or the power ferrite coefficients **Cm**, **X**, **Y**, and **Kdc** to be derived from a manufacturer-provided core loss curve.

Note	The Kdc core loss quantity is not used for the Eddy Current solution type.

4. If you selected **Electrical Steel**, do the following:
   a. Select **Core Loss Coefficient** from the **Calculate Properties for** pull-down list at the bottom of the dialog box.
      The **BP Curve** dialog box appears.
   b. Do one of the following to specify a BP curve:
      - Import the curve from a saved file.
      - Enter the coordinates manually.

Note	The accuracy in inputting the data for B-P Curve for the electrical steel material has significant effect on the correctness of the analyses to the electromagnetic devices. You should input the data for B-P Curve according to accurate data provided by material manufacturers. Typically core material suppliers provide the average loss over a cycle for a peak B, of sinusoidal nature. Therefore for BP curve input in Maxwell, B(Tesla) should be peak and P should be average.

c. Select the units in which the B-P curve is measured from the **Core Loss Unit** pull-down list.

d. Type values and select units for the following:

- **Mass Density**
- **Frequency**
- **Thickness**
- **Conductivity**

As the input data changes, the following parameters are dynamically updated with both the specified unit and the standard unit (w/m^3):

- **Kh** - Hysteresis Coefficient
- **Kc** - Classical Eddy Coefficient
- **Ke** - Excess Coefficient
- **Kdc** - Coefficient Considering DC Bias Effects (not used for Eddy Current solution type)

e. Click **OK**.
The **View / Edit Material** dialog box reappears. The property values for **Kh**, **Kc**, **Ke**, and **Kdc** are updated as new default values.

5. If you selected **Power Ferrite**, do the following:
   a. Select a type for **Cm**, **X**, **Y**, and **Kdc** from the **Type** column.
   b. Type values for the coefficients **Cm**, **X**, **Y**, and **Kdc** in the **Value** column.
6. Click **OK** to close the **View / Edit Material** dialog box.
7. Click **OK** to close the **Edit Libraries** dialog box.

Related Topics

Core Loss Model for a Maxwell Material
Core Loss Coefficients for Electrical Steel
Core Loss Coefficient Extraction
Setting Core Loss for Transient and Eddy Current Solvers
Calculating Properties Using a Hysteresis Loop in Maxwell

Core Loss Coefficients for Electrical Steel

Under sinusoidal flux conditions, core loss is computed in the frequency domain as follows:

$$P_v = P_h + P_c + P_e = K_h f(B_m)^2 + K_c(fB_m)^2 + K_e(fB_m)^{1.5}$$

For a material with anisotropic core losses, the coefficients Kh, Kc, and Ke have different values in different principal directions, that is, (Kh_x, Kh_y, Kh_z), (Kc_x, Kc_y, Kc_z), and (Ke_x, Ke_y, Ke_z) in (x, y, z) directions, respectively.

When a DC component exists in the flux density, the core loss is modified to the following:

$$P_v = C_{dc} K_h f(B_m)^2 + K_c(fB_m)^2 + K_e(fB_m)^{1.5}$$

where

- **B_m** is the amplitude of the AC flux component,
- **f** is the frequency,
- **K_h** is the hysteresis core loss coefficient,
- **K_c** is the eddy-current core loss coefficient, and
- **K_e** is the excess core loss coefficient.

C_{dc} is computed from the following equation:

$$C_{dc} = 1 + K_{dc} B_{dc}^2$$

where

- **B_{dc}** is the DC flux component, and
- **K_{dc}** is the coefficient considering the DC flux bias effects. The default value of K_{dc} is 0.65, that is, when the input value is 0, K_{dc} will be automatically set to 0.65.

Related Topics

Core Loss Coefficient Extraction

Core Loss Coefficient Extraction

The principles of the computation algorithm for **Kh**, **Kc**, and **Ke** are summarized as below.

Note	Since the manufacturer-provided loss curve is obtained under sinusoidal flux conditions at a given frequency, these coefficients can be derived in the frequency domain.

The iron-core loss without DC flux bias is expressed as the following:

$$P_v = P_h + P_c + P_e = K_h f(B_m)^2 + K_c(fB_m)^2 + K_e(fB_m)^{1.5} = K_1 B_m^2 + K_2 B_m^{1.5}$$

where

$$K_1 = K_h f + K_c f^2$$

$$K_2 = K_2 f^{1.5}$$

The classical eddy-current loss coefficient is calculated directly as

$$K_c = \pi^2 \cdot \sigma \cdot \frac{d^2}{6}$$

where σ is the conductivity and *d* is the thickness of one lamination sheets.
Minimize the quadratic form to obtain K_1 and K_2.

$$f(K_1, K_2) = \Sigma \left[P_{vi} - \left(K_1 B_{mi}^{\ 2} + K_2 B_{mi}^{\ 1.5} \right) \right]^2 = min$$

where P_{vi},B_{mi} – the i-th point of the data on the measured loss characteristics curve.
The other two loss coefficients are obtained as

$$K_h = \left(K_1 - K_c f_0^{\ 2} \right) / f_0$$

$$K_e = K_2 / f_0^{\ 1.5}$$

where f_0 is the testing frequency for **Loss Curve**.

Related Topics

Core Loss Coefficients for Electrical Steel

Importing a BP Curve From a File

For the loss characteristics (B-P Curve), you first set the **Core Loss Model** of the material to **Electrical Steel** or **Power Ferrite** as a material property in the View/Edit Material window. When you do so, this enables the Calculate Properties selection for drop down at the bottom of the window. Select either **Core Loss at One Frequency** or **Core Loss versus Frequency** from the drop down menu to open the B-P Curve window.

For **Core Loss at One Frequency**:

1. In the BP Curve dialog box, click **Import Dataset** to open **Import Dataset**dialog box.
2. Find and select the tab separated points file containing the BP curve data points.
3. Click **Open**.
4. Optionally, click **Swap X-Y Data** to switch the B values and P values if they are in the wrong columns.

For **Core Loss versus Frequency**:

1. In the Core Loss versus Frequency dialog box, if needed, enter the desired **Frequency** and click **Add**.
2. Click **Edit Dataset** for the desired frequency to open the **Edit Dataset** dialog box.
3. Click **Import Dataset** to open **Import Dataset**dialog box.
4. Find and select the tab separated points file containing the BP curve data points.
5. Click **Open**.
6. Optionally, click **Swap X-Y Data** to switch the B values and P values if they are in the wrong columns.

Manually Specifying BP Curve Coordinates

For the loss characteristics (B-P Curve), you first set the **Core Loss Model** of the material to **Electrical Steel** or **Power Ferrite** as a material property in the **View/Edit Material** window. When you do so, this enables the Calculate Properties selection for drop down at the bottom of the window. Select either **Core Loss at One Frequency** or **Core Loss versus Frequency** from the drop down menu to open the B-P Curve window.

Select **Core Loss Coefficient** from the **Calculate Properties for** pull-down list to open the **BP Curve** dialog box.

Do the following in the **BP Curve** dialog box:

- Enter **B** and **P** values in each row of the **Coordinates** table.

Note	The value of B must increase along the curve. Placing the cursor in the field in the **Coordinates** table enables the **Add Row** and **Delete Row** buttons.

  As you enter values, the graph is updated.

To add or edit rows, click the following buttons:

- **Add Row Above**
- **Add Row Below**

- **Append Rows** (In the dialog box that appears, specify the number of rows to append to the table, and click **OK**.)
- **Delete Rows**

Calculating Properties for Temperature-dependent Core Loss Curves in Maxwell

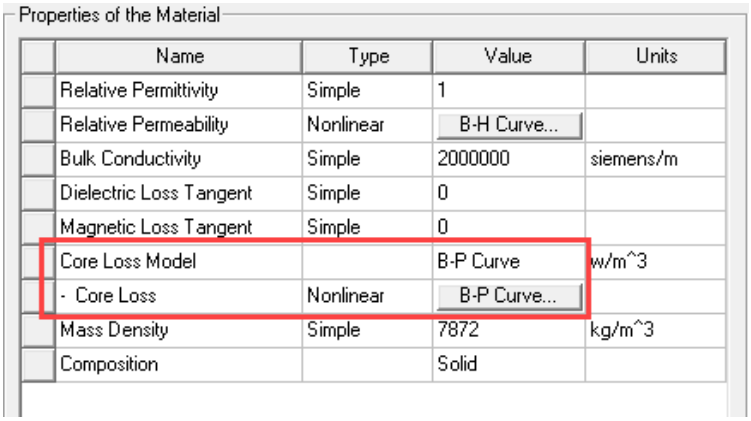
To be able to extract parameters from the temperature-dependent core loss characteristic curves, you first set the **Core Loss Model** of the material to **B-P Curve** as a material property in the **View / EditMaterial** dialog box.

Note	When a material has temperature-dependent core loss curves defined, neither the Thermal Modifier nor the Spatial Modifier are applicable to the material.

To calculate temperature-dependent core loss properties for a material:

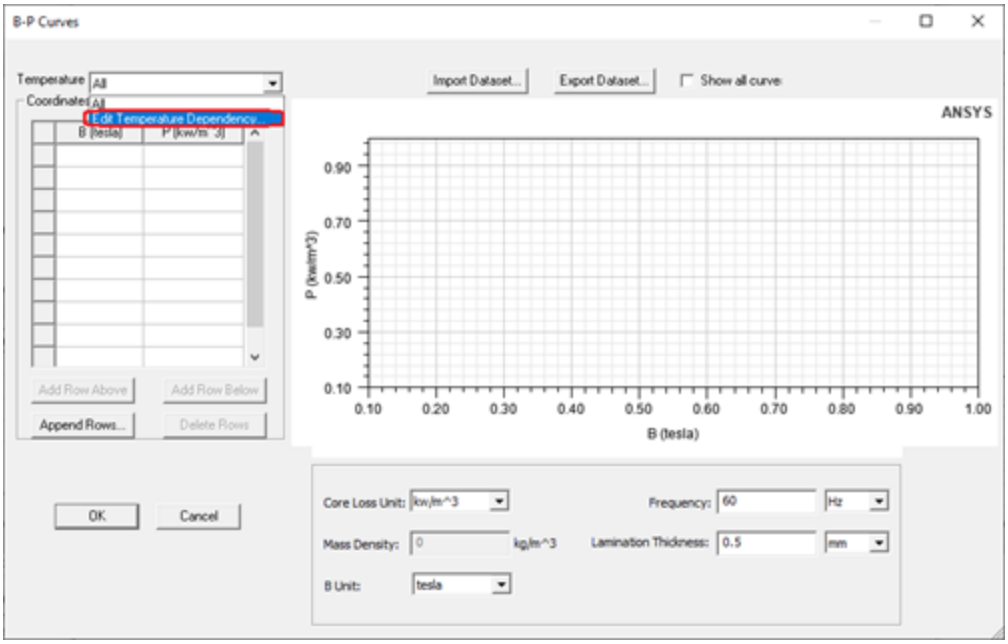
1. Click **Tools>Edit Configured Libraries>Materials**.
   - Alternatively, in the project tree, you can right-click **Materials**, under **Definitions**, and select **Edit All Libraries**.

   The **Edit Libraries** dialog box appears.
2. Click **Add Material**.
   The **View / Edit Material** dialog box appears.
3. In the **Core Loss Model** row, select **B-P Curve** from the **Value** pull-down list.

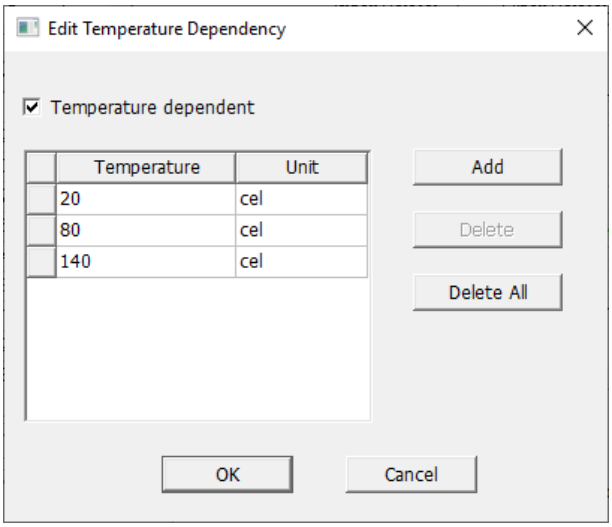


For **Core Loss**, click the **B-P Curve** button.

The **B-P Curve** dialog box appears.



4. From the **Temperature** pull-down list you can do either of the following:
   a. Select **All** to specify a single core loss curve.
   b. To create (or edit) multiple core loss curves, select **Edit Temperature Dependenc**.



   - Click **Add** to add temperatures; then select the desired units. You can also select rows and **Delete** them, or **Delete All**.
   - Check **Temperature dependent** to enable use of the temperature-dependent B-P curves by the solver.
   - Click **OK** to close the dialog box and return to the **B-P Curves** dialog box.
5. Do one of the following to specify a BP curve:

a. Click **Import Dataset** to open the **Import Dataset** dialog box. Find and select the tab separated file containing the B-P curve data points. (Data points must start at the origin.)
b. Enter the coordinates manually. (Data points must start at the origin.)

6. Select the units in which the B-P curve is measured from the **Core Loss Unit** and **B Unit** pull-down lists.
7. Type values and select units for the following parameters:
   - **Mass Density**
   - **Frequency**
   - **Lamination Thickness**
8. Click **OK**.
   The **View / Edit Material** dialog box reappears.
9. Click **OK** to close the **View / Edit Material** dialog box.
10. Click **OK** to close the **Edit Libraries** dialog box.

Related Topics

Core Loss Model for a Maxwell Material

Extracting Equivalent Cut Depth Based on Multiple Curves

The manufacturing effects on core loss computation can be considered in Maxwell transient simulation. To consider the manufacturing effects, you need to input a parameter called **Equivalent Cut Depth**. The default value for this parameter is 1mm, which is close to the average results extracted from some measurements.

A tool is featured to extract this parameter. If you have multiple core loss curves varying with flux density under different number of cut edges (two cut edges are produced by each cut) measured by the single sheet tester method, you can use this tool to extract the parameter.

To be able to use this feature, you must first set the **Core Loss Model** of the material.

To extract the equivalent cut depth:

1. Click **Tools>Edit Configured Libraries>Materials**. The **Edit Libraries** dialog box appears.
   - Alternatively, in the project tree, you can right-click **Materials**, under **Definitions**, and select **Edit Libraries**. The **Edit Libraries** dialog box appears.
   - Or select an object, and click **Modeler>Assign Materials**. The **Select Definition** dialog box appears.
2. Click **View / Edit Material**.
   The **View / Edit Material** dialog box appears.
3. Select **Electrical Steel**, or **Power Ferrite**, or **Hysteresis Model**, from the **Core Loss Model** drop-down menu. The **Equiv. Cut Depth** property is available under **Core Loss Model**, and the **Extract Equiv. Cut Depth** is added in the **Calculate Properties for** drop-down menu.

4. Select **Extract Equiv. Cut Depth** from the **Calculate Properties for** drop-down menu. The **Extract Equiv. Cut Depth** dialog box appears. Click **Add** to add a new row for a loss curve input. Enter a value for number of cut-edges in the **Number** column, and enter the loss curve data in the **Edit Dataset** column.
5. Click the **Edit Dataset…** button. The **Edit Dataset** dialog box appears. Enter values for loss curve **P** and the sinusoidal flux density **B**. Click **OK** to return to the **Extract Equiv. Cut Depth** dialog box .
6. Select the type of curve you want to plot. You can plot for **Selected Number** or **All Numbers**, and plot the **Original Curve** or the **Regression Curve**. To identify the equivalent cut depth, you need to input various measured loss curves with different number of cut edges.
7. Select the **Core Loss Unit**.
8. Type a value for **Mass Density**. This field can only be edited when **Core Loss Unit** is w/lb or w/kg.
9. Enter a value for the **Band Width**. As the input data changes, the parameters are dynamically updated with both the specified unit and the standard unit (w/kg).
10. Click **OK**.

The **View/ Edit Material** dialog box reappears. The extracted values are updated as new default values.

11. Click **OK** to close the **View / Edit Material** dialog box.
12. Click **OK** to close the **Select Definition** or **Edit Libraries** dialog box.

Related Topics

Single Sheet Tester Method

Core Loss Model for a Maxwell Material

Core Loss Coefficients for Electrical Steel

Core Loss Coefficient Extraction

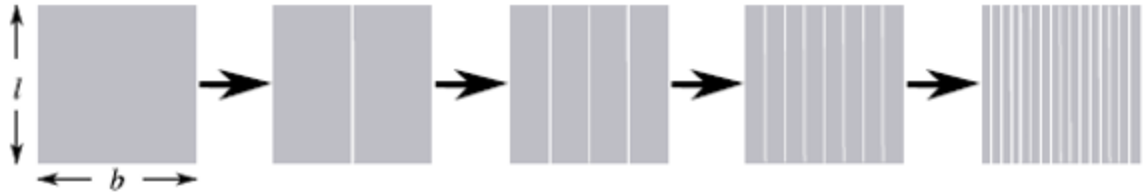
Setting Core Loss for Transient and Eddy Current Solvers

Calculating Properties Using a Hysteresis Loop in Maxwell

Principle of Parameter Identification based on Single Sheet Tester

The device for single sheet tester is shown below:

The single-sheet specimen to be measured with different numbers of cut edges is shown below:



At each cut-edge number N_j, we can measure a loss curve P_{ij} varying with the amplitudes of sinusoidal flux density B_i. Then equivalent cut depth is identified by minimizing the total error between the predicted and measured losses.

$$\varepsilon = \sum_{i=1}^{n}\sum_{j=1}^{m}(P_{ij}'-P_{ij})^2$$

where:

- P_{ij}' and P_{ij} are predicted and measured losses for all points, respectively,
- m is the number of measured loss curves,
- n is the number of points for each loss curve.

In a single sheet tester, the predicted losses can be derived from:

$$P_{ij}' = P_{i0} + \Delta P_{ij} = P_{i0} \cdot (1 + w_e N_j / b)$$

where:

- P_{i0} are the intact losses, the loss curve without cut-edge effects,
- ΔP_{ij} are the increased losses due to the cut-edge effects,
- w_e is Equiv. Cut Depth to be identified, and
- b is the original band width of the single sheet.

In general, the intact losses P_{i0} are unknown, and can be determined by:

$$\frac{\partial \varepsilon}{\partial P_{i0}} = \frac{\partial}{\partial P_{i0}} \left(\sum_{i=1}^{n} \sum_{j=1}^{m} [P_{i0}(1 + w_e N_j / b) - P_{ij}]^2 \right) = 0$$

based on which we derive:

$$P_{i0} = \frac{\sum_{j=1}^{m} P_{ij}(1 + w_e N_j / b)}{\sum_{j=1}^{m} (1 + w_e N_j / b)^2}$$

Similarly, based on:

$$\frac{\partial \varepsilon}{\partial w_e} = 0$$

we get:

$$w_e = b \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} (P_{ij} - P_{i0}) P_{i0} N_j}{\sum_{i=1}^{n} \sum_{j=1}^{m} (P_{i0} N_j)^2}$$

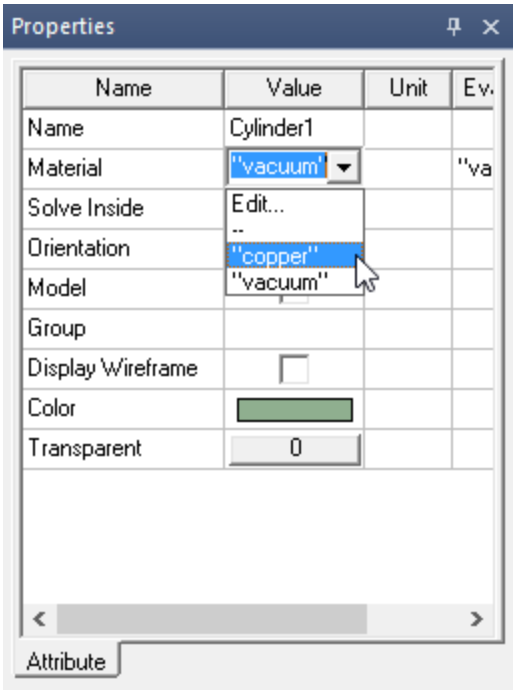
In above equations, intact losses P_{i0} depend on parameter w_e, which in turn depends on the intact losses. An iteration process is required to get the solution. Therefore, to identify the equivalent cut depth, and show the regression curves, we need to input various measured loss curves P_{ij} at different number of cut edges N_j, as well as the band width of the single sheet b.

Related Topics

Core Loss Model for a Maxwell Material
Core Loss Coefficients for Electrical Steel
Core Loss Coefficient Extraction
Setting Core Loss for Transient and Eddy Current Solvers
Calculating Properties Using a Hysteresis Loop in Maxwell

Assigning Materials from the Object Properties Window

The Properties dialog for each object includes a materials property. If you click on the current material property you see a drop-down list that includes an Edit command and a list of materials in the current project. You can select from the list of current materials to assign the selected material to that object.



Related Topics

Assigning Materials
Auto-Complete for Variables and Properties in Electronics Desktop

Viewing and Modifying Material Attributes

1. In the **Select Definition** dialog box or the **Edit Libraries** dialog box, select the material you want to view or modify.

Note	Maxwell's global material library cannot be modified. Materials from the library can be modified, but not saved back to the system library. They can be saved to a User Library or Personal Library.

2. Click **View/Edit Materials**.

   The **View/Edit Material** dialog box appears. The material name and its property values are listed.

Note	By default, not all of the available properties are displayed in the materials table. Only the properties commonly used by the product are displayed. To view the complete table of properties, see Filtering Materials.

3. Under **View/Edit Material for**, select:
   - **Active Design** to display properties used in the active design.
   - **Active Project** to display properties used by this project.
   - **All Properties** to display all properties available. This enlarges the table of properties to show all properties possible. You can use the scroll bars or size the dialog to see all properties.

     Selecting **All Properties** also enables the three **Physics** check boxes: **Electromagnetic**, **Thermal**, and **Structural**. These check boxes are used to filter the properties displayed in the table by physics type. Clearing a check box hides properties of that physics type.
4. You can modify the material as follows:
   a. Provide a new name for the material in the **Material Name** text box.
   b. Under **Type**, specify whether a material property is **Simple,Anisotropic**, **Vector** and **Vector Mag**, or for Relative Permeability, **Nonlinear**, as required for that property.

      For **Simple**, you provide a value or variable.

      For **Anisotropic**, you provide tensor values.

      For **Vector**, you provide a **Vector Mag**, or **Unit Vector** X, Y, and Z Component values.

      For **Non-Linear**, you provide a Data Set.
   c. Provide new material property values in the **Value** boxes.
   d. Change the units for a material property.
5. For **Material Appearance**, you can check the **Use Material Appearance** box to enable the fields for you to specify a color and transparency for the material.
   - Clicking the **Color** bar opens a standard color selection window.
   - Clicking the **Transparency** box opens a **Transparency** dialog with a text field and slider bar for selection.
6. If you want to add descriptive notes for the new material, click the ellipsis button [**...**] by the **Notes** field. This opens a dialog in which you can enter text. To enter multiple lines of notes,

use **CTRL-Enter** to start a new line. (**Enter** by itself will save the Notes field and close the dialog.)

7. Click **OK** to save the changes and return to the **Select Definition** or **Edit Libraries** dialog box.

Warning	If you modify a material that is assigned in the active project after generating a solution, the solution becomes invalid.

Related Topics

Validating Materials
Defining Variable Material Properties
Copying Materials
Removing Materials
Export Materials to a Library
Sorting Materials
Filtering Materials
Working with Materials Libraries

Setting the Material Threshold

The material thresholds tell the **Maxwell 2D** and **Maxwell 3D** solvers how to deal with conductors and insulators. Materials with conductivities above the **Perfect Conductor** threshold are treated as having infinite conductivity and surface current only. Materials with conductivity values below the **Insulator/Conductor** threshold are treated as insulators with no conductivity and no current carrying capability. Materials that fall between the two thresholds are treated as normal conductors that can carry current throughout the volume of the material.

1. Click **Maxwell>Design Settings.** The **Design Settings** dialog box appears. The material thresholds are set in the **Material Thresholds** tab.
2. Type a value in the **Perfect Conductor** text box.
3. Type a value in the **Insulator/Conductor** text box.
4. Click **OK**.

Validating Materials

Maxwell and RMxprt can validate a material's property parameters for an Ansys Electromagnetics software product. For example, it checks if the range of values specified for each material property is reasonable.

If a material's property parameters are invalid, an error message will appear in the lower-right corner of the **View/Edit Material** window. If the parameters are valid, a green check mark will appear there.

To validate the material attributes listed in the **View/Edit Material** window:

- Select a name from the **Properties of the Material** list, and then click **Validate Material**.

Copying Materials

1. In the **Select Definition** dialog box or the **Edit Libraries** dialog box, select the material you want to copy.
2. Click **Clone Material(s)**.

   The **View/Edit Material** dialog box appears.
3. To modify the material's attributes, follow the directions for modifying materials.
4. Click **OK** to save the copy in the active project's material library.

Removing Materials

1. In the **Select Definition** dialog box or the **Edit Libraries** dialog box, select a material you want to remove from the active project's material library.
2. Click **Remove Material(s)**.
   A message appears asking if you want to continue.
3. Click **Yes**.
   The material is deleted from the project material library.

Note	The following materials cannot be deleted: • Materials stored in Ansys Electromagnetics global material library. • Materials that have been assigned to objects in the active project.

In a project library, you may want to use the **Project>Remove Unused Definitions** command to remove selected materials definitions that your project does not require.

Exporting Materials to a Library

1. In the **Select Definition** dialog box or the **Edit Libraries** dialog box, select the material you want to export.
2. Click **Export to Library**.

   The **Save As** file browser appears.
3. Do one of the following to specify the location:
   - Click **PersonalLib** to export the material to a local project directory, accessible only to the user that created it.
   - Click **UserLib** to export the material to a library that is shared by more than one user, usually in a central location.
4. Type the library's file name.
5. Click **Save**.

Sorting Materials

You can change the order of the materials listed in the **Select Definition** dialog box or the **Edit Libraries** dialog box. You can sort the list of materials by name, library location, or material property value.

To change the order of the listed materials:

- Click the column heading by which you want to order the materials.

If the arrow in the column heading points up, the material data are listed in ascending order (1 to 9, A to Z) based on the values in the column you chose. If you want the material data to be listed in descending order (9 to 1, Z to A), click the column heading again. The arrow then points down.

Note	By default, not all of the available properties are displayed in the materials table. Only the properties commonly used by the product are displayed. To view the complete table of properties, see Filtering Materials.

The **Material Filters** tab of the **Edit Libraries** dialog box allows you to specify the default for whether to filter for **This Product** or **All Products**.

Filtering Materials

If you want to remove certain materials or material properties from the list in the **Select Definition** window, use the filter options under the **Material Filters** tab.

You can filter out materials based on the product or library with which they are associated. You can also filter out material properties and types of material properties. And you can remove the filtering in order to see all available material properties.

To filter materials or material properties listed in the **Select Definition** window, using the choices in the **Materials** tab:

1. The field under **Libraries** lists the libraries for the project. Selecting the listed library highlights it and cause the table to display the materials in that library.
2. Above the Libraries area, you can check or uncheck boxes to show or hide Project definitions and all libraries.
   - With both unchecked, nothing appears in the materials table. With both checked, the table shows all materials and highlights those used in the project.
   - With only Project Definitions checked, the materials table shows only the materials used in the project.
   - With All Libraries checked, the table displays all materials, but may not show all available properties.

To filter out or show additional material properties in the **Materials** tab:

1. Click the **Material Filters** tab.
2. Under **Show Materials and Properties for**, select one of the radio buttons:
   - **Active Design** to display properties used in the active design.
   - **Active Project** to display properties used by the current project.

- **All Materials and Properties** to display all available materials and properties. Selecting this enlarges the table of properties shown under the **Materials** tab to show all properties possible. You can use the scroll bars or size the dialog to see all properties.

3. Under **Show Material Types**, click **Select All** to select all of the products listed. Click **Clear** to clear all product selections.
4. Click the **Materials** tab to save your selections.

   Click **Cancel** to revert back to the last saved selections.

Related Topics

Validating Materials
Copying Materials
Removing Materials
Export Materials to a Library
Sorting Materials
Working with Materials Libraries

Working with Material Libraries

There are two different kinds of materials libraries in Maxwell, a system library and a user library.

Related Topics

Editing Libraries

Working with the System Material Library

The library files that ship with Maxwell are stored under the **syslib** directory. These libraries are intended to be read-only and should not be modified. They are available for any material assignment in any project.

Working with User Material Libraries

In addition to the system libraries, Maxwell recognizes two user-configurable library structures: the User Library and the Personal Library. These are used to add user (or company) defined materials. Typically, **userlib** is a network repository for proprietary or corporate definitions available to all seats in an enterprise. **personalLib** contains project and design-specific libraries as needed by individual designs.

A root library directory is set up at installation. If none is specified, the default is the root Maxwell directory.

Materials from libraries in both the User Library and Personal Library, as well as the System Library, are available for use in projects.

Managing Library Files

You can use **Manage Files** to change the names of libraries in the UserLib and PersonalLib directories; move libraries to another location in the directories specified for UserLib or PersonalLib, or delete libraries. In addition, you can change the names of sub-directories of the UserLib and PersonalLib directories, move directories to another location in the directories specified for UserLib or PersonalLib, or delete directories.

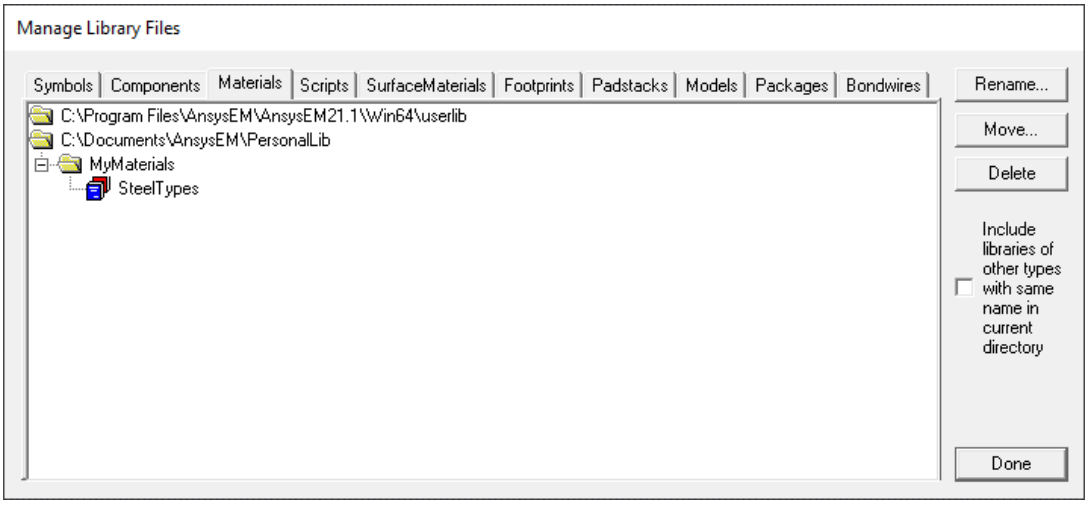
> **Note:**
>
> - You cannot move libraries out of the UserLib or PersonalLib hierarchies.
> - To move a library or directory to a subfolder, that subfolder must already exist.

Materials, components, and other definitions in libraries directly affected by an action which refer to definitions changing library name or location will have those references adjusted. Definitions in libraries indirectly affected – that is, those not having their names or locations modified by an action – with references to definitions in directly affected libraries, are updated.

To manage library files:

1. Click **Tools >Library Tools >Manage Files**.

   The **Manage Library Files** dialog opens on the Components tab. There is a tab for each library type: Symbols, Components, Materials, Scripts, Surface Materials, Footprints, Padstacks, Models, Packages, and Bondwires.



2. Click the tab you want.
3. Select an individual library or a directory on which you want to perform the action.
4. Click one of the following buttons:
    - Rename
    - Move
    - Delete
5. Select the **Include libraries of other types with same name in current directory** check box to cause the action to be performed on other types with the same name in the same directory. For example, if you select “MyProbes” on the Components tab, the action is extended to a symbol library called “MyProbes.aslb” in the same directory.
6. Click **Done** when finished managing libraries.

Related Topics

General Options:Directories

Editing Libraries

There are two different methods of editing libraries.

- Using right-click on Materials in the project window and selecting the **Edit Library** shortcut to display the **Edit Libraries** window.

  Editing definitions from the project window does not modify the configured libraries for any particular design, since this is editing in general.
- Using **Tools>Edit Libraries>Materials** from the menu bar takes the current design into account and adds any new libraries to the configured list for the design.

Related Topics

Adding New Materials

Editing Material Definitions

When editing material definitions or adding a material to the user/personal library, use the menu command (**Tools>Edit Libraries>Materials**) rather than the project tree. Editing or defining definitions by right-clicking in the project tree window does not modify the configured libraries for any particular design – it only edits the material in general. When you use the menu command, the current design is considered, and any new materials are made available to assign in that current design. New or modified materials can also be exported to a library in the User Library or Personal Library from the **Edit Libraries** dialog.

To edit a material's definition:

1. Click **Tools>Edit Libraries>Materials**.

   The **Edit Libraries** dialog appears.
2. In the **Search Parameters** section under **Libraries**, select the library database. By default the **[sys] Materials** system library is selected.
3. Select the material you want to edit, and click **View/Edit Materials**.

   The **View/Edit Material** dialog appears.
4. Make the desired changes to the **Material Name**, the **Material Coordinate System Type**, and the properties listed in the **Properties of the Material** section.
5. From the **View/Edit Material for** section, select whether the changes shall apply to the (**Active Design**, **Active Project**, or **All Properties**).
6. You may optionally set a *Thermal Modifier* to specify that certain material properties have characteristics that vary with temperature.
7. Click **Validate Material** to make sure your changes are valid.

   If the material properties are valid, a green check mark appears below the **Validate Material** button.
8. Optionally, use the **Calculate Properties for...** drop down list to calculate properties for a permanent magnet or a Non-Linear Permanent Magnet.
9. Optionally, to calculate properties for core loss, select **Core Loss Coefficient** from the **Calculate Properties for...** drop down list.
10. When you are done making changes to the material properties, click **OK**.
    The **Edit Libraries** dialog reappears.
11. If you are finished in the **View/Edit Material** dialog box, click **OK** to close it.

Related Topics

Auto-Complete for Variables and Properties in Electronics Desktop

Setting a Thermal Modifier

Materials used in 3D Designs, and for sheet objects in 2D Designs, may have a Thermal Modifier assigned to change specific material properties with temperature, primarily for use when coupling to thermal analysis.

To specify Thermal modifiers for a material:

1. In the View/ Edit materials dialog you must enable the View/ Edit Modifier check box for Thermal Modifier.

   This causes the Properties of the Material table to expand to include a Thermal Modifier column. By default, the Thermal Modifier property is set to **None**.
2. Selecting **Edit...** from the drop down menu, rather than **None** causes the **Edit Thermal Modifier** dialog to appear.
3. Select the **Expression** radio button to display the **Parameters Modifier** text field or the **Quadratic** radio button to display the tabs for **Basic Coefficient Set** and **Advanced Coefficient Set**.
   - With **Expression** selected, you can write an equation for a thermal modifier in the Parameters **Modifier** text field.

     Checking **Use temperature dependent data set** disables the Modifier text field. You can then use the drop down menu to select Add/Import Dataset. This lets you define the thermal modifier as a data set.
   - With the **Quadratic** radio button selected, in the **Basic Coefficient Set** tab, you can edit fields for the TempRef and units, and fields for C1 and C2 for the following equation:

     $$P(Temp) = Pref[1+ C1(Temp - TempRef) + C2(Temp - TempRef)^2]$$

     where the Pref is defined as the reference relative permittivity.
   - With the **Quadratic** radio button selected, in the **Advanced Coefficient Set** tab, you can edit fields for lower and upper temperature limits (TL and TU respectively) and select their units from the drop down.

     You can also edit the constant value limit for the thermal modifier values outside the limits. By default, these are automatically calculated. Uncheck **Auto Calculate TML, TMU** to specify new values for thermal modifier lower (TML) and thermal modifier upper (TMU) limits.
4. Click **OK** to accept the edits and return to the View/ Edit materials dialog.

Note	Thermal modifier for permeability is disabled when temperature-dependent BH curves are specified.

Related Topics

Adding Datasets

View/ Edit materials dialog

Auto-Complete for Variables and Properties in Electronics Desktop

Set Ambient Temperature

Irreversible Demagnetization Due to Temperature Change

Core Loss Model for a Maxwell Material

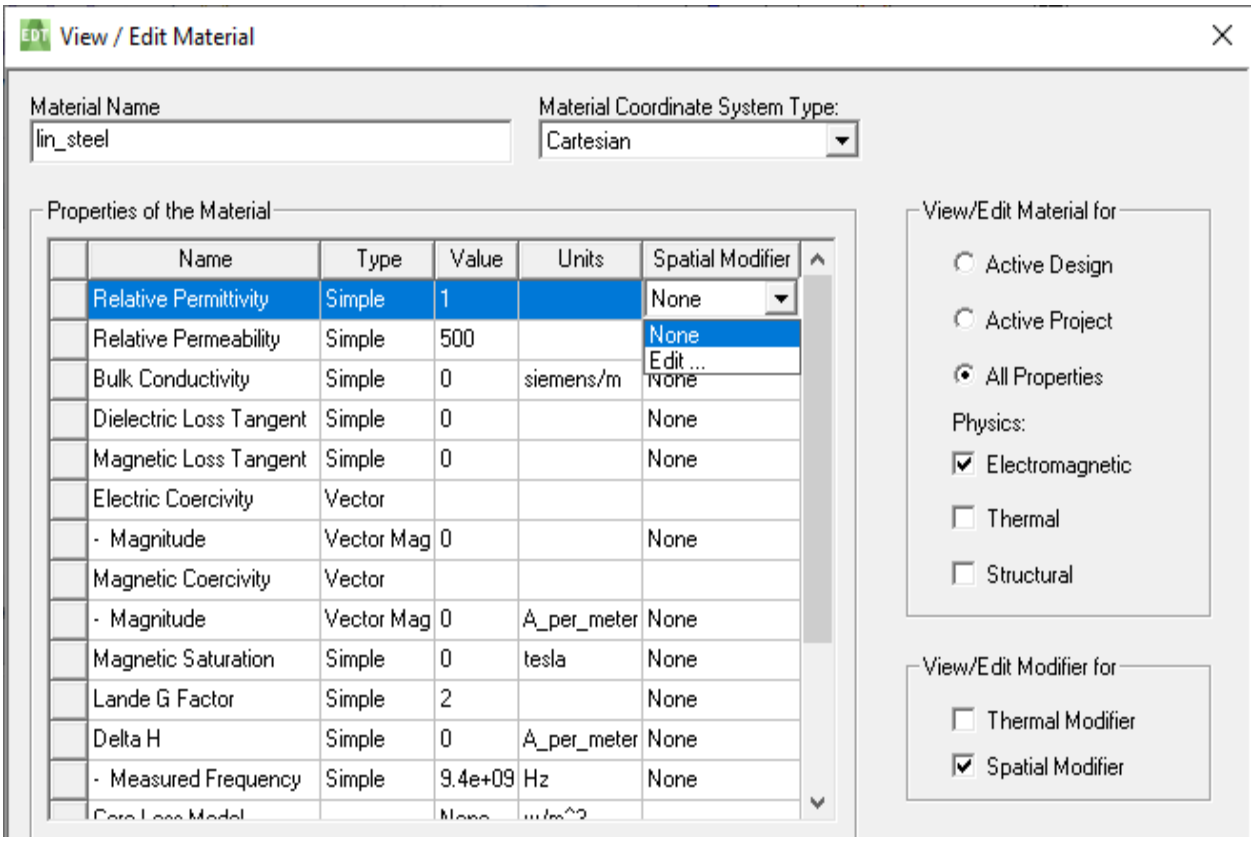
Setting the Temperature of Objects

Setting a Spatial Modifier

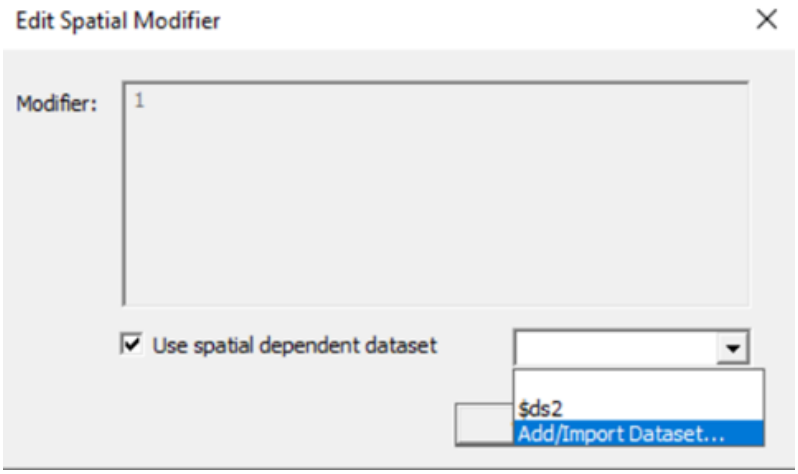
Maxwell magnetic transient solvers support use of a 3D Dataset as a Spatial Modifier to modify spatially varying properties of conductivity, permeability, and permittivity. In general, Thermal Modifier and Spatial modifier cannot be applied to one material property simultaneously.

To specify 3D dataset as Spatial modifiers for a material:

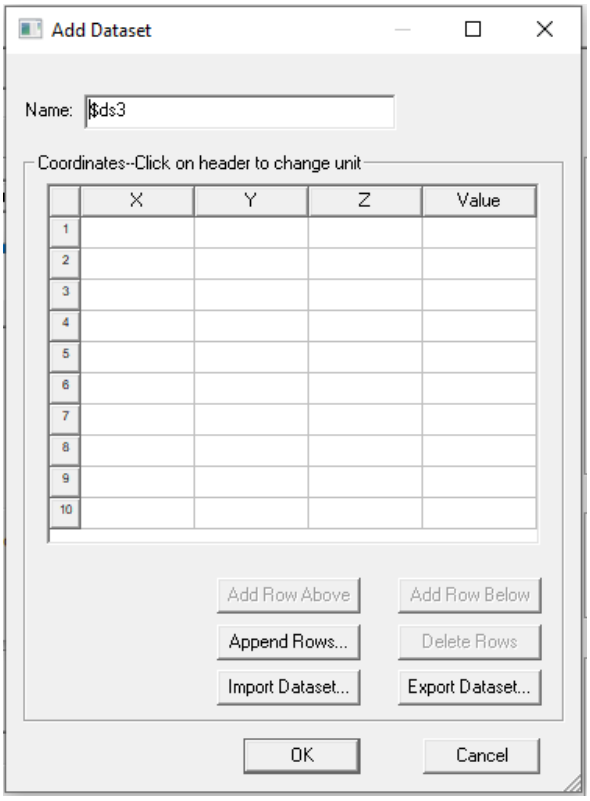
1. In the View/Edit materials dialog you must enable the **Spatial Modifier** check box in the **View/Edit Modifier for** panel. This causes the Properties of the Material table to expand to include a **Spatial Modifier** column. By default, the **Spatial Modifier** property is set to **None**.



2. Selecting **Edit...** from the dropdown menu causes the **Edit Spatial Modifier** dialog to appear.



3. Check **Use spatial dependent dataset**. If a predefined dataset is available, you can select it. Or you can select **Add/Import Dataset**. This opens the **Add Dataset** dialog box for specifying 3D object coordinates and associated values of the material property (conductivity, permeability, or permittivity) to be used in the design. (You can click the column heads to select the units.)



You can also import a tab delimited text file containing predefined coordinates and property values.

Imported datasets must be tab-delimited files (*.tab) containing sets of X, Y, and Z coordinates and associated property Values. For example:

```
2 4 4 68
2 5 5 70
3 5 5 71
```

You can also edit existing datasets as needed.

4. The spatial modifier expression models the material property change with respect to location along the object. The expression appears in the Spatial Modifier column for the associated property.

   The expression is of the form: `clp(dataset_name,X,Y,Z)`

   where `clp` is the closest point interpolation function, and *dataset_name* is the name of the design dataset containing the set of 3D object coordinates and associated property values to be used in the design. For example: `clp(ds2,X,Y,Z)` is the expression that uses the `ds2` dataset.

Related Topics

Defining Functional Material Properties

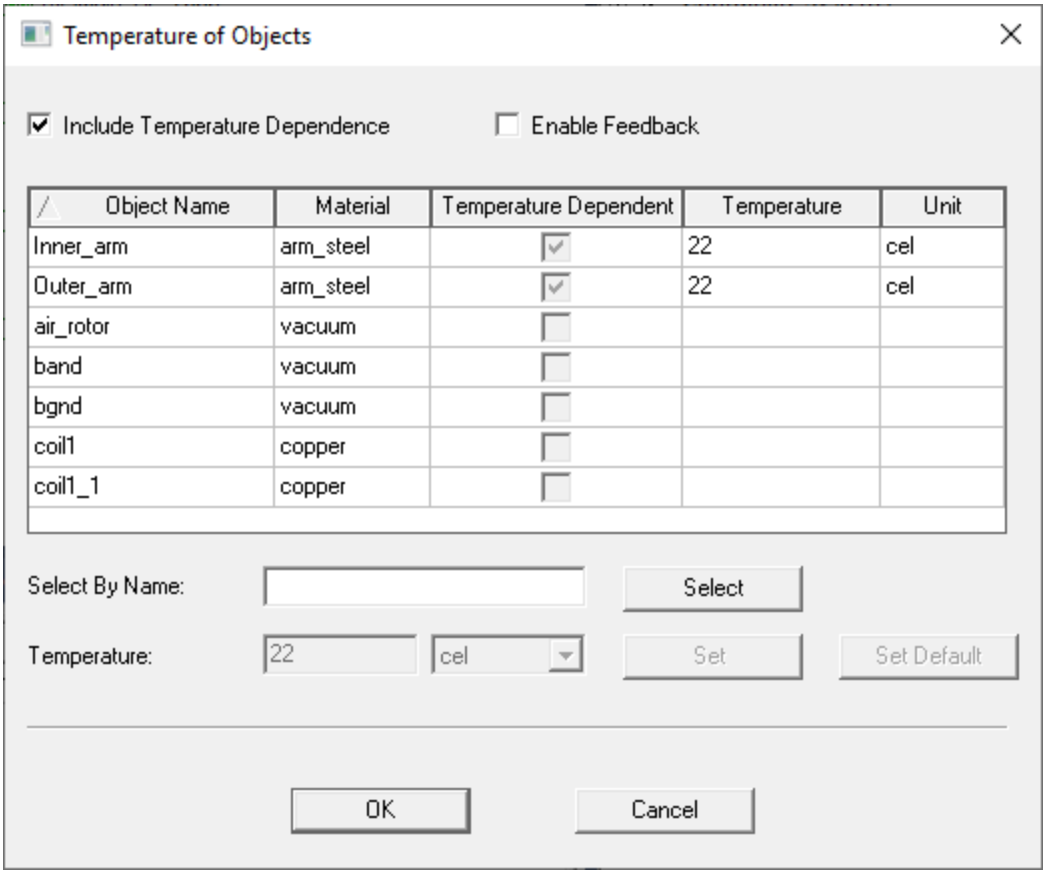
Adding 3D Datasets

Setting the Temperature of Objects

To set the temperature of temperature dependent objects:

1. Use the **Maxwell>Set Object Temperature** command to display the **Temperature of Objects** dialog box.
   - For all design types the dialog includes a table of the objects in the design. It lists the **Object Name**, the **Material** assigned to the object, a non-editable **Temperature Dependent** check box to show whether that object has temperature dependent features (such as those specified by *Setting a Thermal Modifier*). For temperature dependent items, the remaining columns give the associated **Temperature** and **Unit**

of measure.



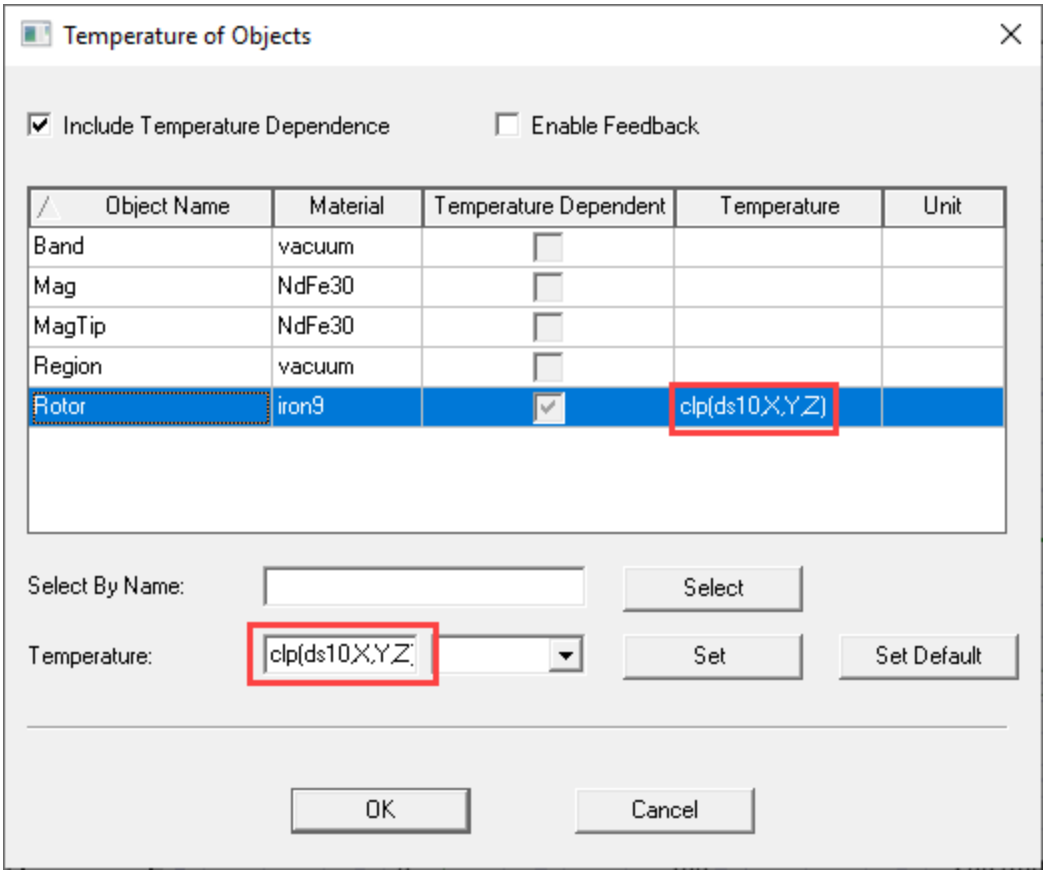
2. Maxwell allows you to enable/disable temperature dependency independently for each design in the same project. To include temperature dependency and enable the editing features, check **Include Temperature Dependence**. This makes the table rows editable.
    - The headers for each of the columns – except the **Temperature Dependent** column – include sort direction capability. You can invert the sort direction in each column by clicking the header. If the list is longer than the display, you can use a scroll bar on the right of the table.
    - In the **Select By Name** field, you can enter the name of the object (or boundary for 3D eddy current designs) you want to select, then click **Select** to highlight the selected row. You can make multiple selections.
3. To set the initial temperature for selected **Temperature Dependent** objects (or boundaries):
    a. Type a value, an existing variable name, or a *spatial modifier expression* in the **Temperature** text field; then select the unit of measure from the drop down menu.

    Refer to step 4 for detailed information on using a *spatial modifier expression*.

b. Click **Set** to apply the value to the selected temperature dependent objects (or boundaries); or click **Set Default** to make the specified values the default. (If you click **Set Default**, the rows for the selected objects (or boundaries) display the Temperature value and units.)

Note	You can also edit the **Temperature** and **Unit** for temperature dependent objects (or boundaries) directly in the table.

4. *Optionally*, for 2D and 3D transient designs, you can enter a *spatial modifier* expression in the **Temperature** field in the **Temperature of Objects** dialog as shown in the figure below. The spatial modifier expression models the temperature material property change with respect to location along the object. You can enter the expression either in the Temperature column at the top, or the Temperature text edit field at the bottom.

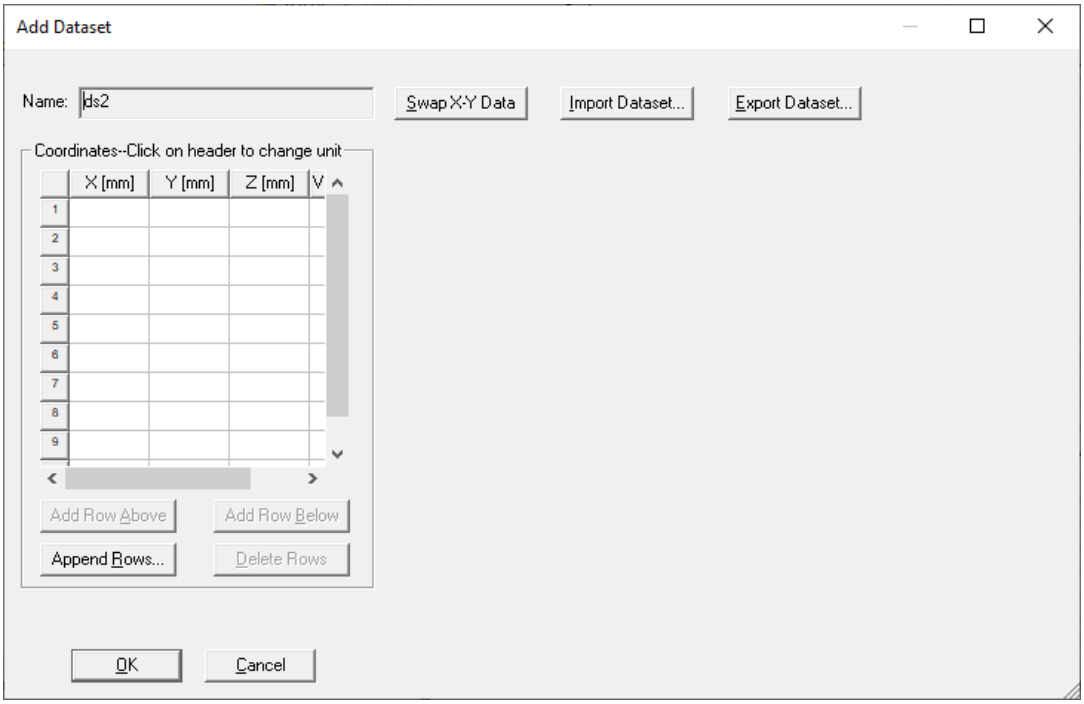


The expression is of the form: `clp(`*`dataset_name`*`,X,Y,Z)`

where `clp` is the closest point interpolation function, and *dataset_name* is the name of the design dataset containing the set of 3D object coordinates and associated temperatures to be used in the design. For example: `clp(ds2,X,Y,Z)` is the expression that uses the `ds2` dataset.

If the expression contains an undefined dataset, when you click **OK** and close the **Temperature of Objects** dialog box, an **Add Dataset** dialog for creating a new design

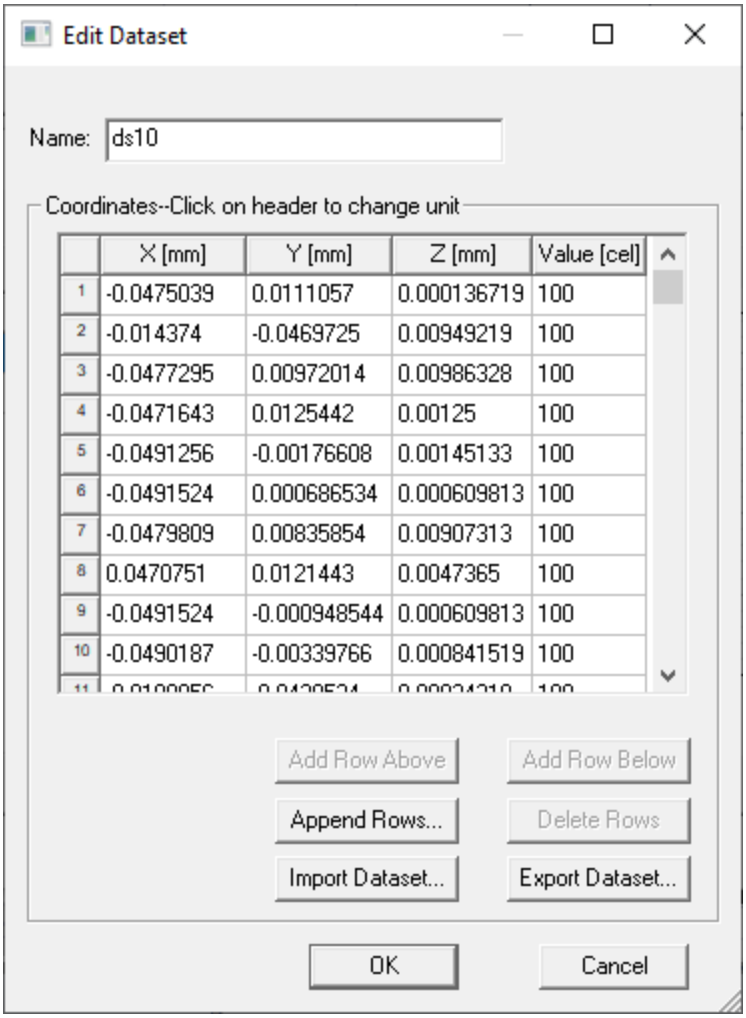
dataset will pop up for inputting or importing the spatial data. You can click the column heads to select the units.



Imported datasets must be tab-delimited files (*.tab) containing sets of X, Y, and Z coordinates and associated temperature Values. For example:

```
2 4 4 68
2 5 5 70
3 5 5 71
```

You can edit existing datasets as needed.



5. When **Include Temperature Dependence** is enabled, you can edit the material for an object (or boundary) by clicking on the material in the row to display a drop down menu listing the current material, and an **Edit...** menu item.

   Select **Edit**... to display the Materials dialog in which you can edit the current material – or select a different material for the object (or boundary). The material editor also allows you to set a thermal modifier. When finished, close the dialog to return to the Temperature of Objects dialog box. If you set a thermal modifier, the Temperature Dependent column for that object (or boundary) will be checked.
6. The **Enable Feedback** check box appears when Maxwell is used with the Ansys Workbench. If checked, you can perform thermal analysis based on a Maxwell solution.
7. To close the **Temperature of Objects** dialog and accept the changes, click **OK**.

Related Topics

Reverting Objects to Initial Temperature

Setting a Thermal Modifier

Core Loss Model for a Maxwell Material

Reverting Objects to Initial Temperature

When a design is enabled for thermal feedback, a **Revert to Initial Temperature** menu item is added to the context menu of the solution setup item and analysis setup folder in the project tree. This menu item is enabled when there are exported thermal data from Mechanical or when thermal data has been incorporated into the current Maxwell solution. Choosing **Revert to Initial Temperature** reverts the temperatures of objects to their initial temperatures.

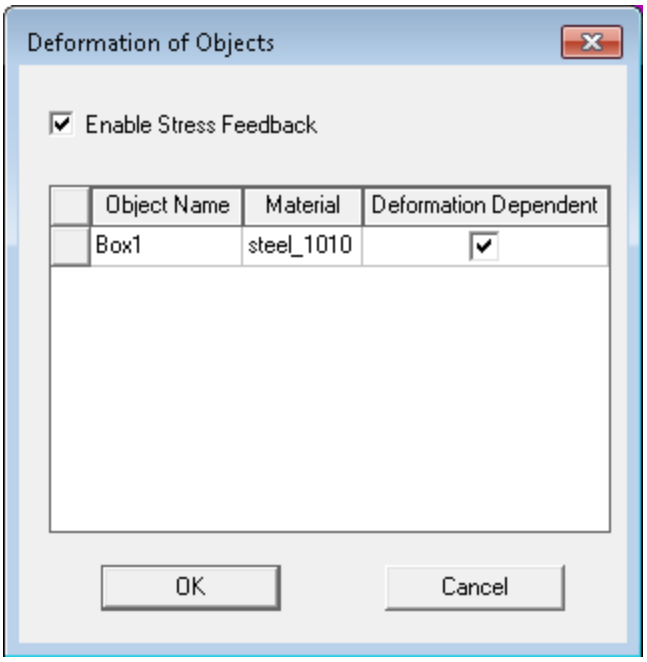
Setting Deformation of Objects for Stress Feedback

For Maxwell systems that are coupled to Ansys Structural via Workbench., users can enable objects for stress feedback in the **Deformation of Objects** dialog box.

To set the deformation of stress dependent objects:

1. Use the **Maxwell>Deformation of Objects** command to display the **Deformation of Objects** dialog box.

   This dialog box includes a table of objects in the design. The first column lists the **Object Name**. The second column lists the **Material**. The **Deformation Dependent** column displays a check box to show whether that object has deformation-dependency enabled.



   Objects without an assigned material, non-model objects, non-3D objects in Maxwell 3D designs, and non-2D objects in Maxwell 2D designs cannot be handled by the solver and thus do not appear in the table.

2. To include deformation dependency, check **Enable Stress Feedback**.

This makes the table objects selectable. The headers for the **Object Name** column and the **Material** column include sort direction arrows. You can invert the sort direction in each column by clicking the header. If the list is longer than the display, you can use a scroll bar on the right of the table.

3. Check the **Deformation Dependent** check box for those objects you wish to include in the stress feedback analysis.

Note	Switching the state of **Enable Stress Feedback** or **Deformation Dependent** for any object invalidates the existing solution. Existing solutions include all field and non-field solutions, such as force and torque. When these options are switched from on to off, existing field solutions include previously incorporated displacement data. On the next simulation, the design will re-solve the last solved pass.

4. To close the **Deformation of Objects** dialog box and accept the changes, click **OK**.

Related Topics

Reverting Objects to Zero Displacement

Reverting Objects to Zero Displacement

When a design is enabled for stress feedback, a **Revert to Zero Displacement** menu item is added to the context menu of the solution setup item and analysis setup folder in the project tree. This menu item is enabled when there are exported displacement data from structural or when displacement data has been incorporated into the current Maxwell solution. Choosing **Revert to Zero Displacement** reverts the displacement values of objects to zero.

Permanent Magnets

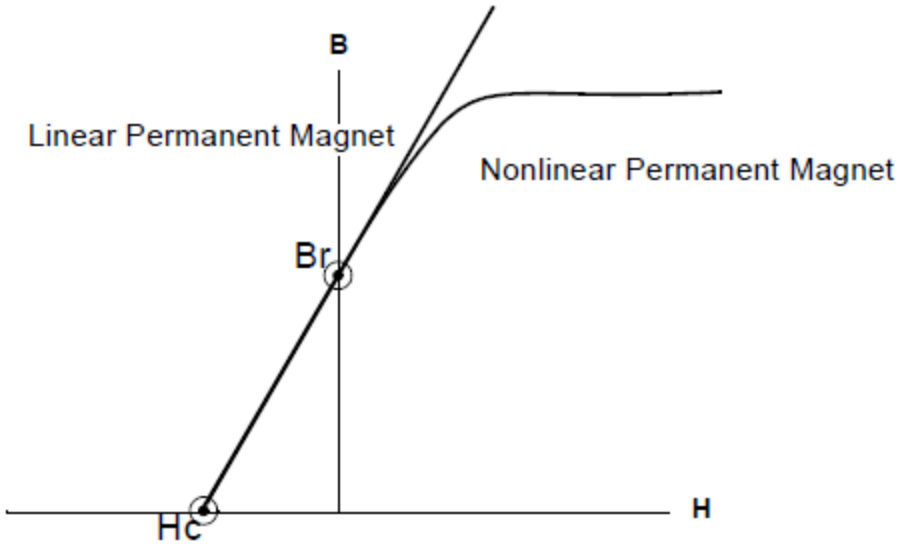
A permanent magnet is defined as a material that generates a magnetic flux due to permanent magnetic dipoles in that material.

Related Topics

Non-Linear vs. Linear Permanent Magnets
Characteristics and Main Parameters of Permanent-magnetic Materials
Calculating the Properties for a Permanent Magnet
Calculating the Properties for a Non-Linear Permanent Magnet
Calculating Properties Using a Hysteresis Loop in Maxwell

Nonlinear vs. Linear Permanent Magnets

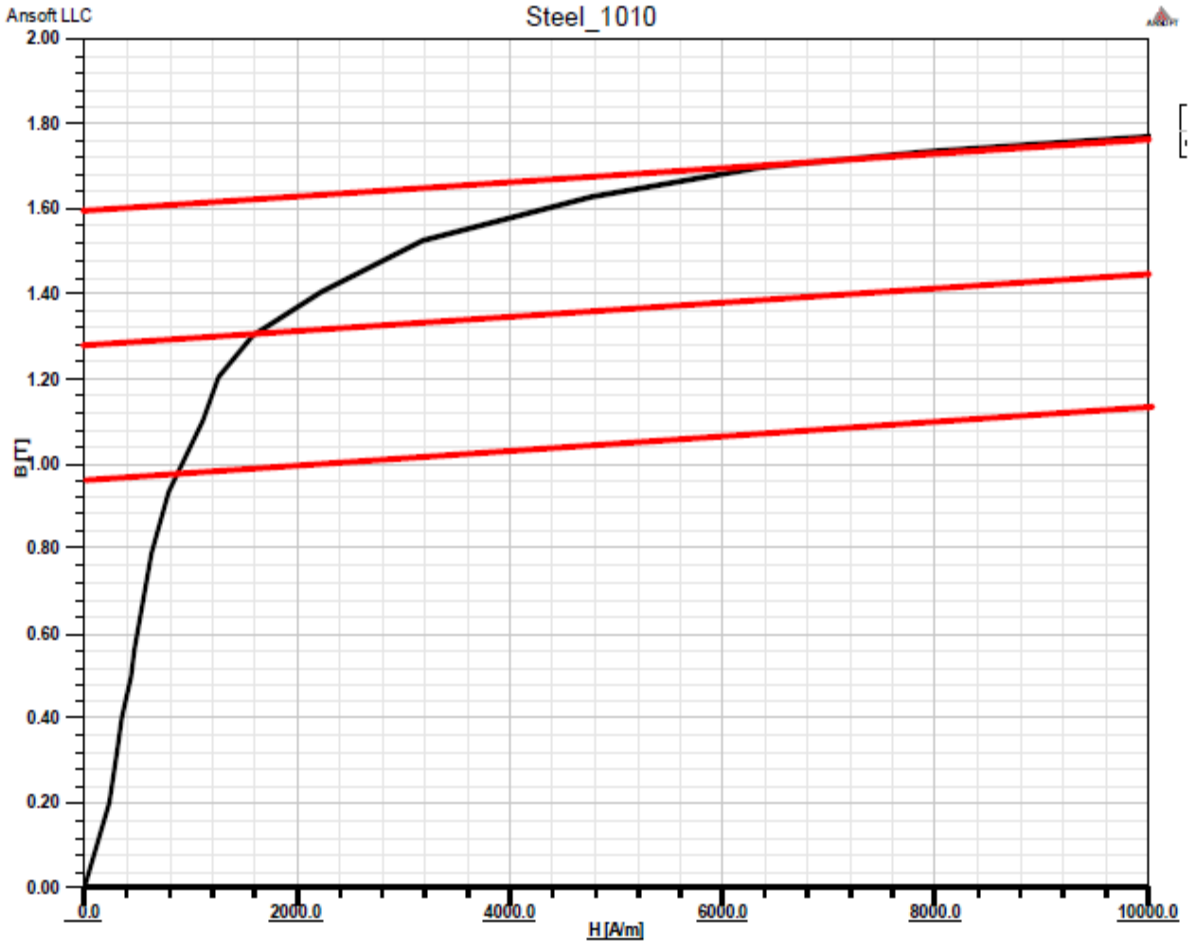
In general, permanent magnets are nonlinear and should be modeled via a B-H curve as shown below. The magnetic coercivity, H_c, is defined as the B-H curve's H-axis intercept, and the magnetic remanence, B_r, as its B-axis intercept.



In many applications, however, the permanent magnet's behavior can be approximated using a linear relationship between **B** and **H**. In these cases, there is no need to create a nonlinear material. Simply enter the appropriate values of B_r or H_c for the material when defining its properties.

Compute Remanent Br from B-H curve

The value of the remanent Br of the individual element after the magnetization field is computed is determined in such a way: after having located the operating point on the original non-remanent B-H curve, draw a line which is parallel to the original recoil curve with the slope of $\mu_0\mu_r$ and passes the operating point, the intersection of this line with B-axis is the remanent Br as the result of the applied magnetization field.



Temperature Dependent Nonlinear Permanent Magnets

To consider the temperature dependence of the demagnetization behavior, the demagnetization curve is described by temperature dependent parameters, which can be derived from supplier datasheets. For a better representation of any types of magnets by a function, it is advantageous to work with an intrinsic flux density $\boldsymbol{B_i}$ versus $\boldsymbol{H}$ curve, instead of a flux density $\boldsymbol{B}$ vs $\boldsymbol{H}$ curve. The relationship between $\boldsymbol{B_i}$ and $\boldsymbol{B}$ is:

$B = B_i + \mu_0 H$	(1)

The temperature dependency of an intrinsic BH curve can be specified by two temperature dependent parameters: remanent flux density B_r and intrinsic coercivity H_{ci}. Here B_r is the value of B_i (and B) when H=0 and H_{ci} is the value of H at B=0. Both B_r and H_{ci} can be described using second order polynomials of temperature T as:

$$B_r(T) = B_r(T_0)(1 + \alpha_1(T - T_0) + \alpha_2(T - T_0)^2) = B_r(T_0)P(T) \quad (1)$$

and

$$H_{ci}(T) = H_{ci}(T_0)(1 + \beta_1(T - T_0) + \beta_2(T - T_0)^2) = H_{ci}(T_0)Q(T) \quad (1)$$

where T_0 is the reference temperature, and α_1, α_2, β_1, and β_2 are coefficients which can be identified from supplier datasheets. In general, the thermal modifiers *P(T)* and *Q(T)* also can be described by any user-defined function of temperature T, in addition to the above second order polynomials.

Consequently, once a B_i-H curve is constructed by a function using certain techniques, such as curve fitting at the reference temperature T_0, any B_i-H curves at other temperatures can be dynamically reconstructed in terms of the temperature dependence of $B_r(T)$ and $H_{ci}(T)$. Finally, the B-H curve in the second and the third quadrants can be further derived through the conversion using equation (1).

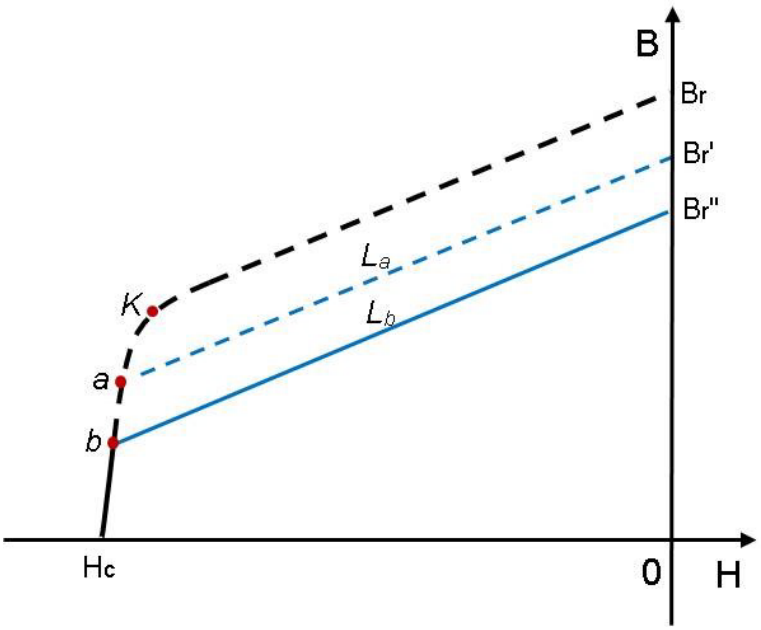
Since the model specifies two temperature dependent parameters: remanent flux density B_r (α) and intrinsic coercivity H_{ci}(β), the thermal modifier *P(T)* is specified in the field associated with the input of relative permeability (or B_i-H curve), and the thermal modifier *Q(T)* is specified in the field associated with the magnitude of coercivity for H_{ci}.

Irreversible Demagnetization Due to Temperature Change

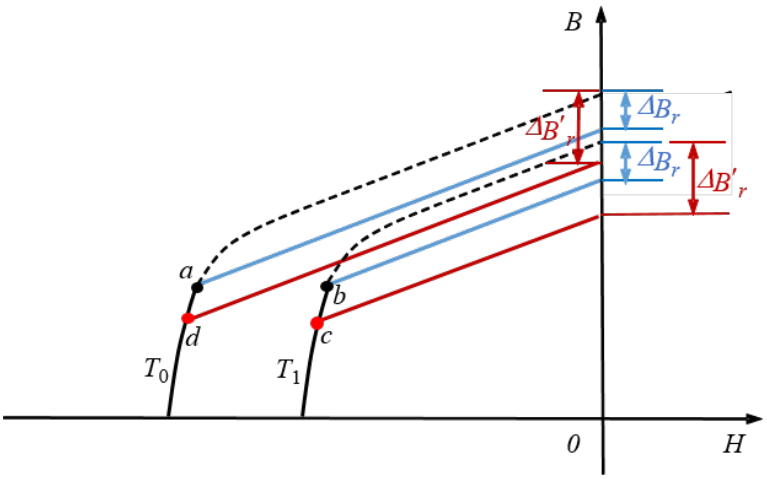
Permanent magnets exposed to heat can lose strength which cannot be recovered. For example, if a permanent magnet motor is operated under a torque overload condition, the motor will draw more current to produce the required torque. The increased current causes some demagnetization, further increasing the current draw. Over a period of time, this overload may cause the motor to overheat. The resulting rise in temperature can lower the magnet material's intrinsic coercivity thus making the magnet prone to demagnetization. If the temperature exceeds the magnet's operating point, some of the magnet's domains become permanently demagnetized.

Consequently, when the magnet's temperature decreases to its initial value, the operating point does not return to its original value. This effect is irreversible.

The following image shows that when a new working point ***a*** goes below the knee point ***K***, the subsequent working point will change along a new constructed B-H curve which consists of two curve segments: the recoil line L_a and the demagnetizing curve part below point ***a***. The residual flux density of the recoil line is *Br'*, and the difference between *Br* and *Br'* is the irreversible *Br* loss, which is denoted as $\Delta Br = Br - Br'$.



When the temperature of a permanent magnet with operating point ***a*** (as shown below) increases from T_0 to T_1, based on user-defined thermal modifiers, a new demagnetization curve can be constructed element by element at the new temperature, T_1. Since the initial irreversible *Br* loss ΔBr remains unchanged, the initial operating point on the new demagnetization curve can be identified as point ***b*** (the intersection of the new demagnetization curve and the recoil line which passes ($Br - \Delta Br$). It follows that the worst-case operating point at temperature T_1 can be further evaluated as indicated by point ***c*** from the field solution. This new worst-case operating point, *c*, causes the irreversible *Br* loss to increase from ΔBr to $\Delta B'r$. If the temperature then changes from T_1 back to T_0, the operating point will not return to point *a*. Instead, it will be located at ***d*** – associated with temperature T_0 based on the new irreversible loss $\Delta B'r$. The Maxwell solution is able to store the design's irreversible demagnetization values for different working temperatures.



Characteristics and Main Parameters of Permanent-magnetic Materials

The permanent-magnetic material belongs to hard-magnetic material. It is characterized with "fat" hysteresis loop, which encloses large area as shown in the figure. When magnetized, it keeps high magnetic property with the external magnetic field removed, therefore is used in the permanent-magnet electric machine to produce magnetic field. The characteristics of the permanent-magnetic material are represented with its main parameters: residual flux density B_r, coercive field force H_c and maximum magnetic energy product $(BH)_{max}$.

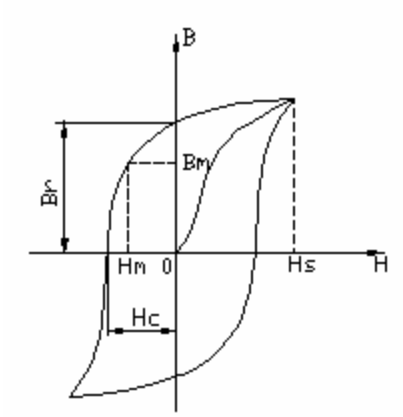


Figure 10-1 "Fat" Hysteresis Loop

Related Topics

Demagnetization Curve

Recoil Lines

Calculating the Properties for a Permanent Magnet

Calculating the Properties for a Non-Linear Permanent Magnet

Calculating Properties Using a Hysteresis Loop in Maxwell
Temperature Dependent Nonlinear Permanent Magnets

Calculating the Properties for a Permanent Magnet

From the V**iew/Edit Materials** window:

1. Set the **Relative Permeability** to **Simple**.
   This enables the **Calculate Properties for...** drop down menu at the bottom of the window.
2. Click **Permanent Magnet** from the drop-down menu.
   This displays the **Properties for Permanent Magnet** dialog box. This contains the following fields.

Mu (enabled by default)	Provide a value for relative permeability.
Hc (enabled by default)	Coercive field force H_c in the units specified. Provide a value and select units from the drop-down menu.
Br/Mp (disabled by default)	Checking this enables the radio buttons that let you specify either Br or Mp.
Br	Residual flux density B_r, in Tesla. If enabled, provide a value and select units from the drop-down menu.
Mp	Permanent Magnetization M_p, in A/m. If enabled, provide a value and select units from the drop-down menu.

3. Click **OK** to close the dialog and return to the **View/Edit Materials** window.
   The values for **Relative Permeability** and **Magnitude** under **Magnetic Coercivity** are updated as new default values.

Related Topics

Non-Linear vs. Linear Permanent Magnets
Characteristics and Main Parameters of Permanent-magnetic Materials
Calculating the Properties for a Non-Linear Permanent Magnet
Temperature Dependent Nonlinear Permanent Magnets
Irreversible Demagnetization Due to Temperature Change

Calculating the Properties for a Nonlinear Permanent Magnet

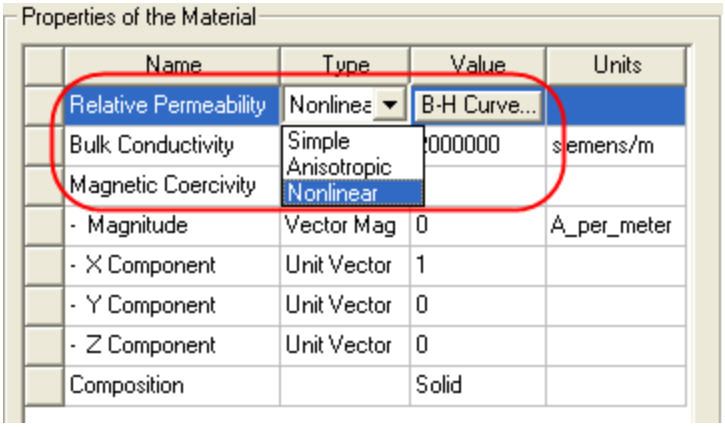
Nonlinear permanent magnet properties may be specified in one of two ways:

- Input a BH Curve directly
- Using the **Properties for Non-Linear Permanent Magnet** dialog box

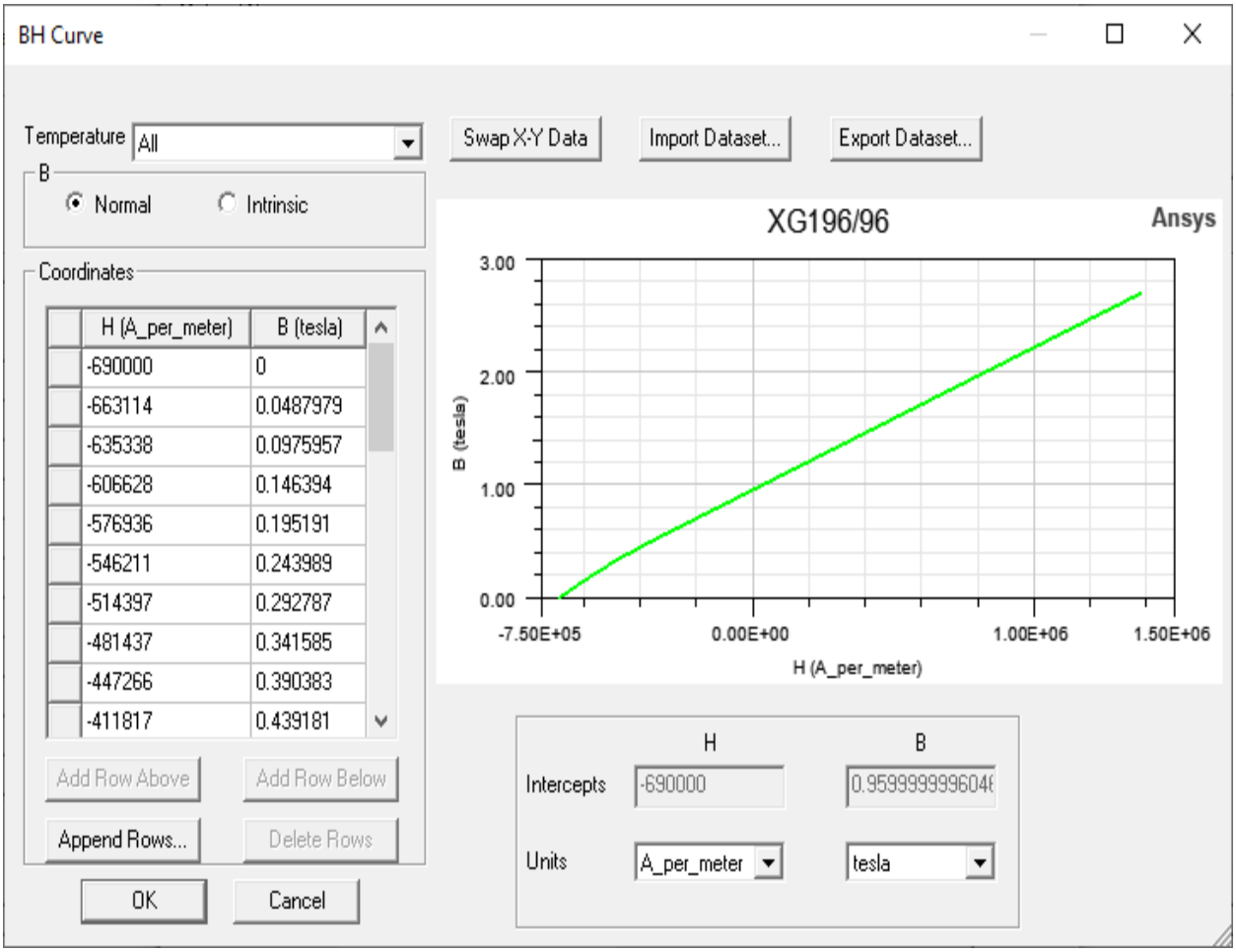
Input a BH curve directly as follows:

> **Note:** You can model multiple temperature dependencies of a nonlinear material property by providing two or more temperature-dependent BH curves. Refer to *Specifying BH Curves for Nonlinear Relative Permeability* for detailed information.

1. Click the **View/Edit Materials…** button in the **Edit Libraries** dialog box.
   The **View/Edit Material** dialog box appears.
2. The nonlinear BH curve is defined by setting the **Relative Permeability** **Type** to **Nonlinear**.



   A **B-H Curve** button appears in the nonlinear property's **Value** column.
3. Input the BH curve by clicking the **B-H Curve** button in the property **Value** column.
   This opens the **BH Curve** dialog box in which you can input (or modify) curve data. (Refer to *Adding Datasets* for general information on working with datasets.



- For a **Normal** BH curve, the slope of the curve can not be less than that of free space anywhere along the curve.
- For an **Intrinsic** BH curve, the slope of the curve can not be less than 0.

Note	• The **Intrinsic** BH curve is supported only in Maxwell 2D/3D magnetostatic and transient design types. A material property defined using an Intrinsic BH curve will fail validation check in all the other product/design types. • When an **Intrinsic** BH curve is added, the **Relative Permeability Value** button label changes to **Bi-H Curve** as visual indication of the type of curve currently defined for the material.

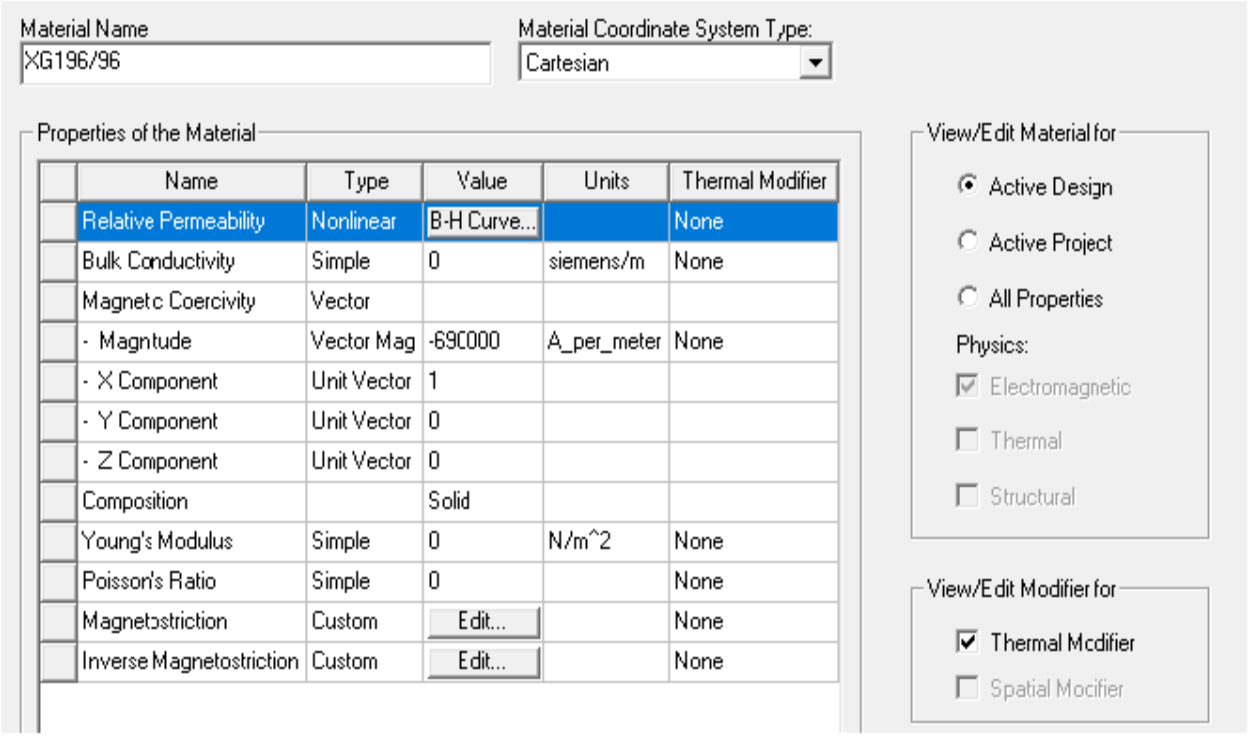
When you **OK** the dialog box, an error message displays if a slope is out of tolerance, identifying the data points between which the slope is less than that of free space.

The operations to input a nonlinear demagnetization curve are the same as entering a BH curve for Steel material. When a BH curve goes through the second quadrant or third quadrant, the curve is treated as a demagnetization curve.

4. To model a single *temperature dependency* for a nonlinear permanent magnet you must:

> **Note:** You can model multiple temperature dependencies of a nonlinear material property by providing two or more temperature-dependent BH curves. Refer to *Specifying BH Curves for Nonlinear Relative Permeability* for detailed information.

- Use an **Intrinsic** BH curve to model the **Relative Permeability**
- Specify a Thermal Modifier for both **Relative Permeability** and the **Magnitude** of **Magnetic Coercivity**. Apply a thermal Modifier by selecting the **Thermal Modifier** check box. Checking this box causes the **Thermal Modifier** column to display at the right side of the **Properties of the Material** table. Selecting **Edit** rather than **None** causes display of the **Edit Thermal Modifier** dialog.



Alternatively, a non-linear BH curve can be modeled by the following four parameters:

- residual flux density B_r
- coercive field force H_c
- maximum energy product $(BH)_{max}$
- relative recoil permeability μ_r

From the **View/Edit Materials** window:

1. Set the **Relative Permeability** to **Nonlinear**.
   This enables the **Calculate Properties for...** drop down menu at the bottom of the window.
2. Click **Non-Linear Permanent Magnet** from the drop down menu.

This displays the **Properties for Non-Linear Permanent Magnet** dialog box, which contains the following fields into which you enter the appropriate values.

Mur	Provide a value for relative permeability.
Hc	Coercive field force H_c in the units specified. Provide a value and select units from the drop down menu.
Br	Residual flux density B_r, in Tesla. If enabled, provide a value and select units from the drop down menu.
BH max	Maximum magnetic energy product $(BH)_{max}$ If enabled, provide a value and select units form the drop down menu.

3. Click **OK** to close the dialog and return to the **View/Edit Materials** window.

   The values for **Relative Permeability** and **Magnitude** under **Magnetic Coercivity** are updated as new default values.

Related Topics

Non-Linear vs. Linear Permanent Magnets

Characteristics and Main Parameters of Permanent-magnetic Materials

Calculating the Properties for a Permanent Magnet

Temperature Dependent Nonlinear Permanent Magnets

Irreversible Demagnetization Due to Temperature Change

Calculating Properties Using a Hysteresis Loop in Maxwell

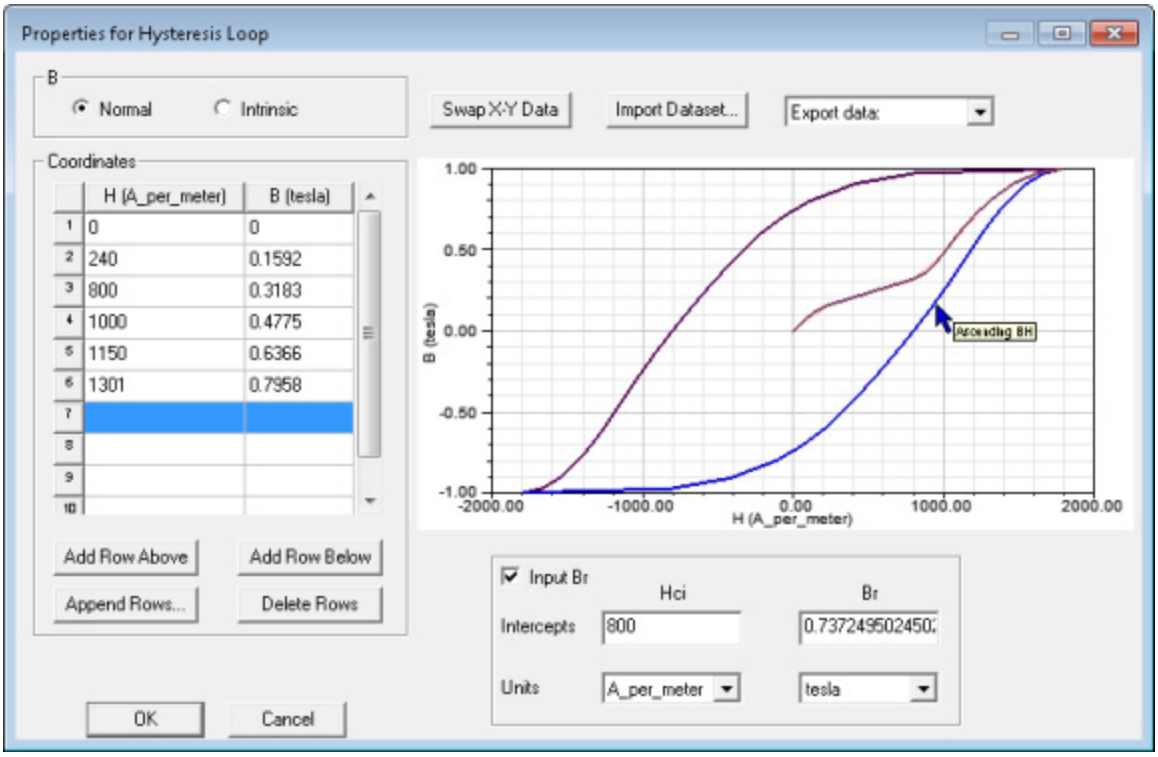
For a material whose Relative Permeability is Nonlinear (refer to *Calculating the Properties for a Non-Linear Permanent Magnet*), and whose **Magnitude** of **Magnetic Coercivity** is zero, you can calculate core loss properties using a hysteresis loop as follows:

1. Click **Tools>Edit Configured Libraries>Materials**. to open the **Edit Libraries** dialog box.
   - Alternatively, in the project tree, you can right-click **Materials**, under **Definitions**, and select **Edit All Libraries**.
2. Click **Add Material** to open the **View/ Edit Material** dialog box.
3. In the **Core Loss Model** row, select **Hysteresis Model** from the **Value** pull-down list.

   If you select either **View/Edit Material for: This Product** or **All Products** radio button, additional parameters appear in the table below **Core Loss Model** (**Intrinsic Coercivity Hci**, **Remanency Br**, **Kc**, and **Equiv. Cut Depth**). You can input these values directly. You can also supply the **Hci** and **Br** values in the **Properties for Hysteresis Loop** dialog box as described below.

   The **Hysteresis Loop** selection in the **Calculate Properties for** pull-down list at the bottom of the dialog box is also enabled. This allows the coefficients **Hci**, and **Br** to be derived from a manufacturer-provided core loss curve.

4. Select **Hysteresis Loop** from the **Calculate Properties for** pull-down list at the bottom of the dialog box to open the **Properties for Hysteresis Loop** dialog box The current BH Curve coordinates are loaded by default. **Hci** and **Br** values from the **Intrinsic Coercivity Hci**, **Remanency Br** fields in the **View / Edit Material** dialog box are also displayed.



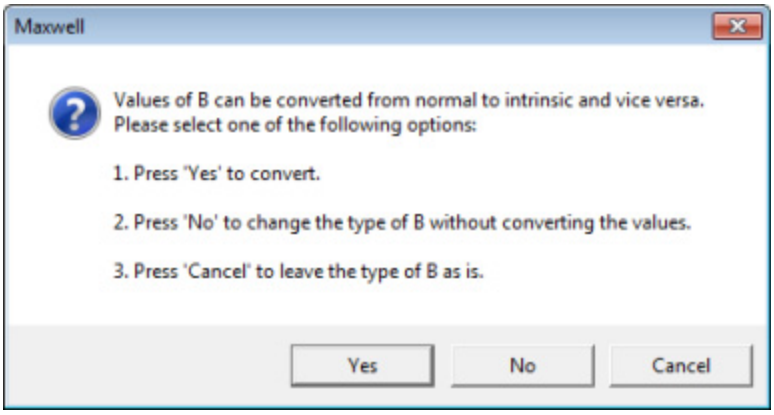
5. Do one of the following to specify the hysteresis loop curve:
   - Enter the **B** and **H** values in each row of the **Coordinates** table manually. The graph updates as you enter values. Buttons allow you Add, Append, and Delete table rows as needed.

Note	• The value of B must increase along the curve. • The accuracy of input data for the curve has significant effect on the correctness of the analyses of the electromagnetic devices. You should ensure curve data provided by material manufacturers is accurate.

   - You can also import the hysteresis loop coordinates from a saved file.
     a. Click **Import Dataset** to open the **Import Dataset** dialog box.
     b. Find and select either the BH Data .bh file, or the tab-delimited .tab file containing the curve data points. Click **Open** to import the data.
     c. Optionally, click **Swap X-Y Data** to switch the B and H values if they are in the wrong columns.
6. Optionally, enter the Hci and Br values and choose appropriate units for them. Input Br must be enabled to enable the Br entry box; otherwise, the default zero value indicates that **Br** will be determined automatically by the software. (These fields may already have values in them if previously supplied in the **View / Edit Material** dialog box.) Disabling Input Br resets

the Br value to zero.

7. Optionally, select either **Normal** (the default) or **Intrinsic** for BH curve input and plot display. When changing from Normal to Intrinsic (or vice-versa) the following dialog displays, allowing you to choose whether or not to convert B values.



8. Optionally, you can export data for the descending and ascending curve of the hysteresis loop. You can also export the BH curve as well as the smoothed BH curve.
9. Click **OK** to close the Properties for Hysteresis Loop dialog.

   The values of Magnetic Coercivity Magnitude, **Hci**, and **Br** are updated in the **View/Edit Material** dialog.

Note	In the **View/Edit Material** dialog box, if the **Magnetic Coercivity Magnitude** is changed to a non-zero value by modifying the **Relative Permeability B-H Curve** property, the **Core Loss Model** property is reset from **Hysteresis Model** to **None**.

10. Click **OK** to close the **View / Edit Material** dialog box.
11. Click **OK** to close the **Edit Libraries** dialog box.

Related Topics

Core Loss Model for a Maxwell Material

Magnetostriction and Inverse Magnetostriction

Magnetostriction is a property of ferromagnetic materials that causes their deformation during the process of magnetization. The structure of such materials is divided into *domains*, each of which is a region of uniform magnetic polarization. When a magnetic field is applied, the domains rotate causing a change in the material's dimensions. This change in the material's dimensions is due to the anisotropic crystalline structure of the material – it takes more energy to magnetize a crystalline material in one direction than another. For example, the steel used for the thin laminations in transformers is manufactured by passing it through a series of rollers to achieve the desired thickness. Consequently its crystalline structure – and associated magnetic domains – differs along the direction in which it was rolled (the *rolling* direction) versus the *transverse* direction.

Magnetostriction – and Inverse Magnetostriction – can be regarded as an energy transduction (or transformation) from magnetic to mechanical and vice-versa. It can also be described as a bidirectional magneto-mechanical coupling between the mechanical and magnetic fields in the magnetostrictive material. It is magnetostriction that causes the low-pitched humming sound that can be heard coming from transformers – caused by the changing magnetic field produced by oscillating AC currents. For detailed information on these properties, refer to *Magnetostriction Modeling of Magnetic Materials* in the technical notes section.

- Magnetostriction/inverse magnetostriction properties can be applied to Maxwell 2D/3D magnetosatic and transient solutions through customizable Magnetostriction and Inverse Magnetostriction material properties.
- Maxwell desktop supplies the Magnetostriction and Inverse Magnetostriction material properties to the Maxwell solver. The solver then calculates force density with magnetostriction force components included.
- For Maxwell Magnetostatic solutions, if any material used has the Inverse Magnetostriction property defined, Maxwell desktop saves the initial H field, which is generated by the solver without exterior stress/strain applied, in the project results.
- Magnetostriction and inverse magnetostriction are applicable to Maxwell 2D/3D Magnetosatic design one-way and two-way coupling with Ansys Mechanical system.

Note	Inverse magnetostriction for materials having cylindrical and spherical coordinate systems is not supported.

Related Topics

Calculating Properties for Magnetostriction

Calculating Properties for Inverse Magnetostriction

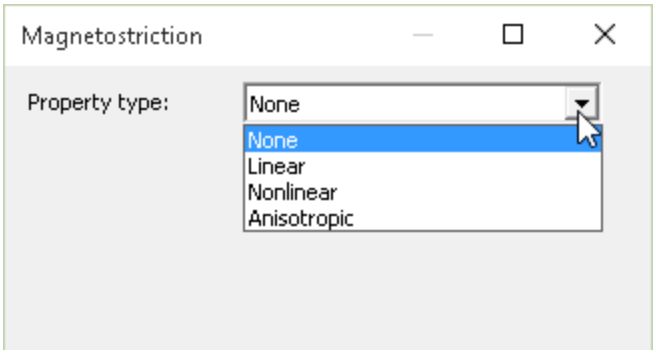
Magnetostriction Modeling of Magnetic Material

Calculating the Properties for Magnetostriction

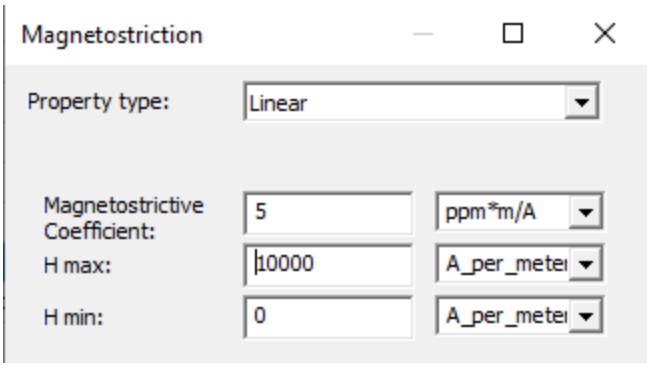
The magnetostriction property supports three models: Linear, Nonlinear, and Anisotropic. The property values you enter typically come from manufacturer data sheets.

To define these model properties, from the **View/Edit Material** dialog:

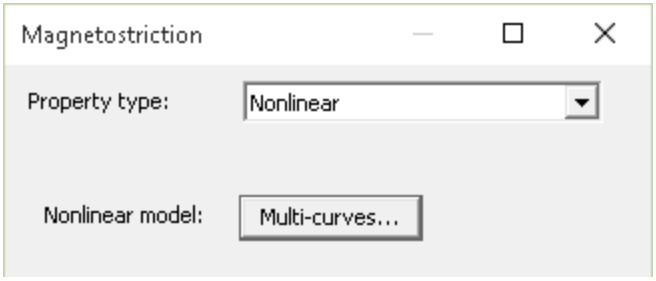
1. Click the **Edit** button to the right of **Magnetostriction Custom** to open the Magnetostriction properties dialog.



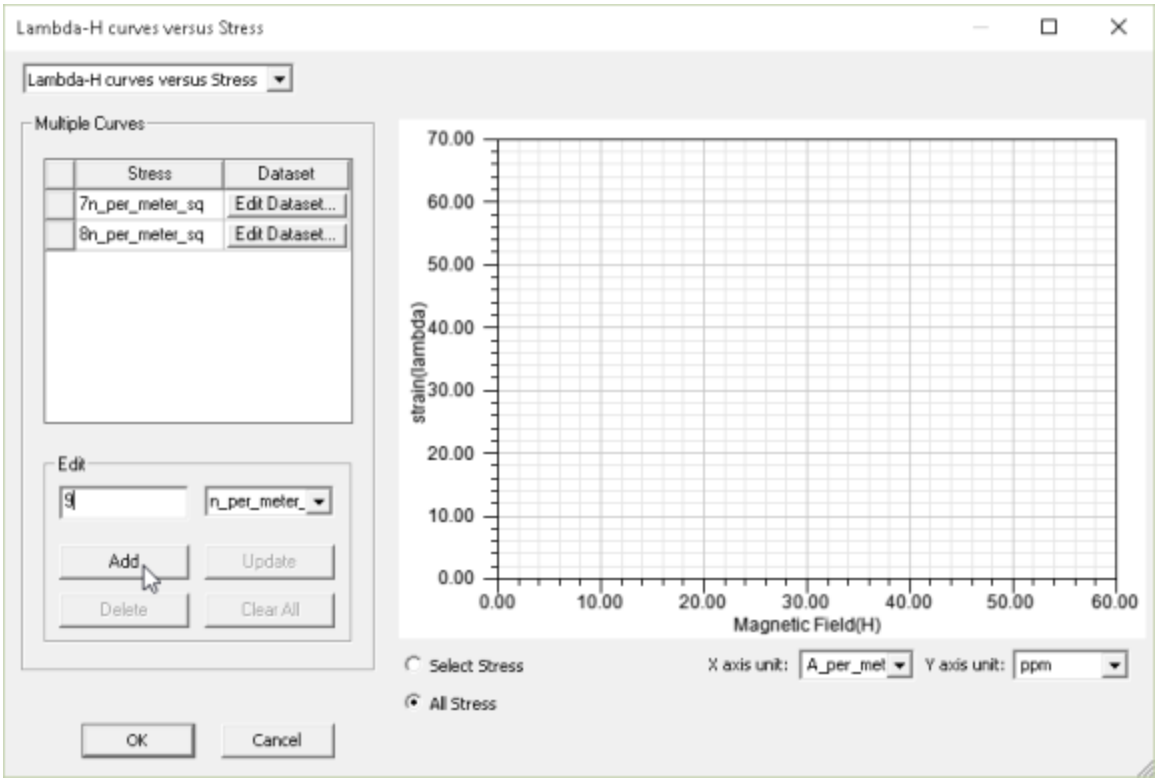
2. Choose a **Property type** from the drop-down menu: **Linear**, **Nonlinear**, or **Anisotropic** and do one of the following:
   a. For **Linear**, enter values for **Magnetostrictive Coefficient**, **H max**, and **H min**.



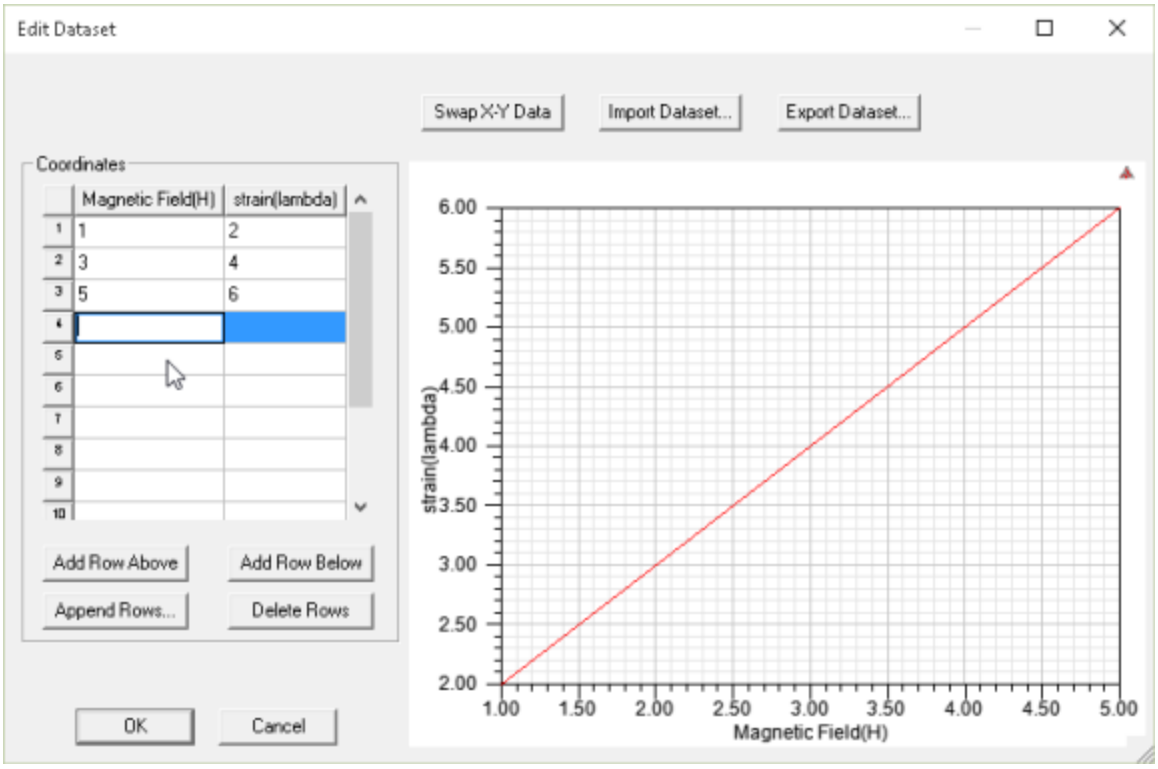
   b. For **Nonlinear**, click the **Multi-curves** button



   to add one or more Stress dependent Lambda-H curves.



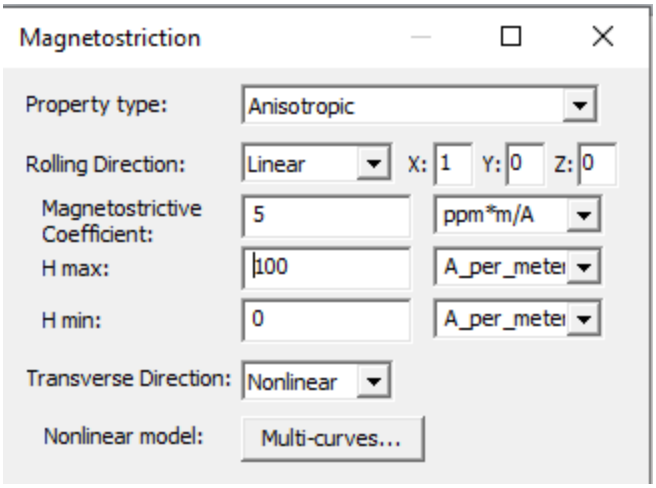
Enter a stress value in the **Edit** text box and click **Add**, then click **Edit Dataset** to open the **Edit Dataset** dialog box, in which you enter coordinate values for the **Magnetic Field (H)** and **strain (Lambda)**.



When finished, click **OK** to close the dialogs and return to the **Magnetostriction** dialog.

c. For **Anisotropic**, select either **Linear** or **Nonlinear** from the **Rolling Direction** and **Transverse Direction** drop-down menus. Rolling Direction refers to the alignment of the crystalline structure (and associated magnetic domains) in the direction along which the material (such as the steel laminations used in transformers) passed through the rollers during manufacture. Transverse Direction is at right angle to the Rolling Direction. Magnetostriction property values typically differ in each direction.

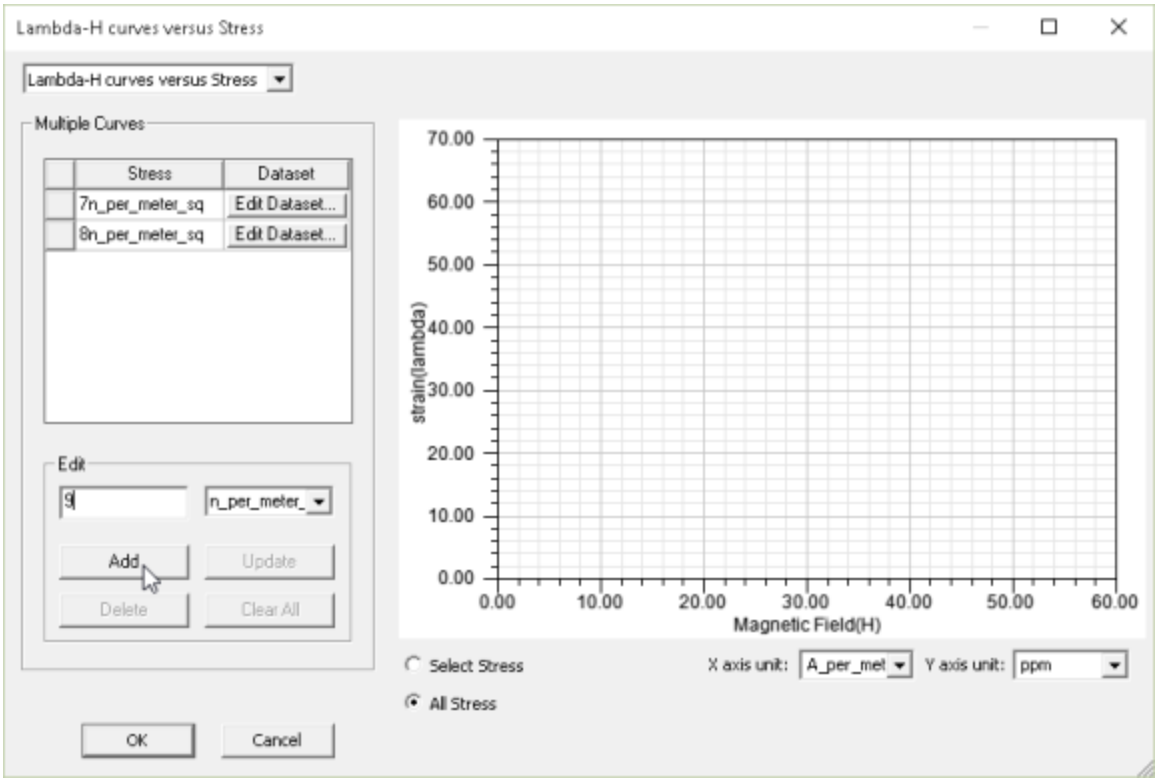
Note	The type you select for each of the directions need not be the same. For example, you could select Linear for Rolling Direction and Nonlinear for the Transverse Direction, and vice versa.



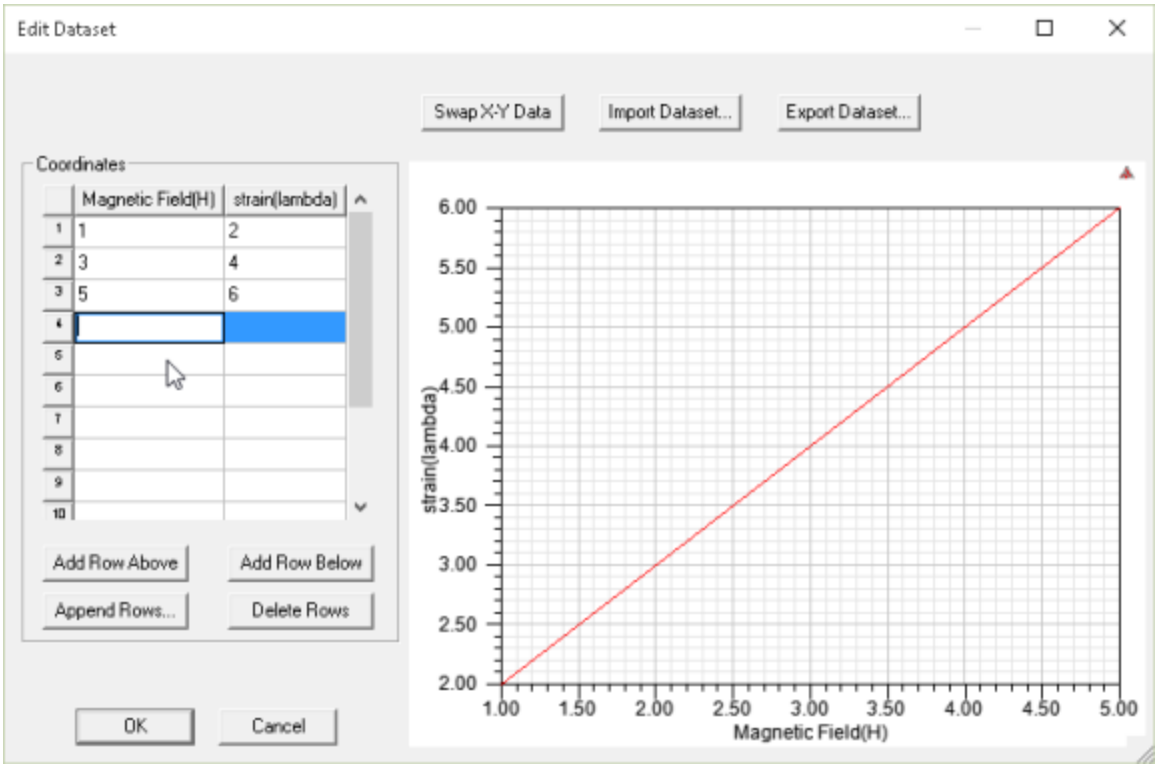
- For **Linear**, enter values for **Magnetostrictive Coefficient**, **H max**, and **H min**; and for **Rolling Direction** only, enter the **X**, **Y**, and **Z** coordinate values.

Note	The X, Y, and Z components of the Rolling Direction correspond to the **Material Coordinate System Type** set in the **View/Edit Material** dialog. • For **Cartesian**: X=X, Y=Y, Z=Z • For **Cylindrical**: X=R, Y=Phi, Z=Z • For **Spherical**: X=Rho, Y=Theta, Z=Phi

- For **Nonlinear**, click the **Multi-curves** button to add one or more Stress dependent Lambda-H Curves.



Enter a stress value in the **Edit** text box and click **Add**, then click **Edit Dataset** to open the **Edit Dataset** dialog box, in which you enter coordinate values for the **Magnetic Field (H)** and **strain (Lambda)**.



When finished, click **OK** to close the dialogs and return to the **Magnetostriction** dialog.

3. Click **OK** to close the **Magnetostriction** dialog and return to the **View/Edit Materials**dialog.

Related Topics

Magnetostriction Modeling of Magnetic Material

Magnetostriction and Inverse Magnetostriction

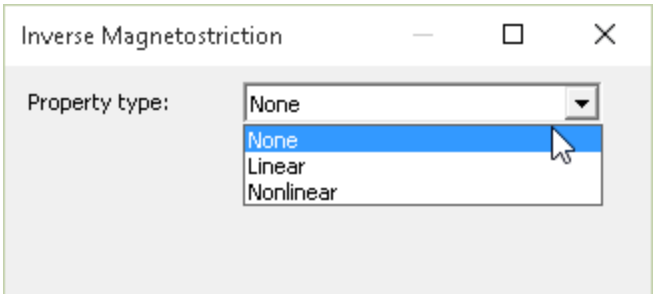
Calculating the Properties for Inverse Magnetostriction

Calculating the Properties for Inverse Magnetostriction

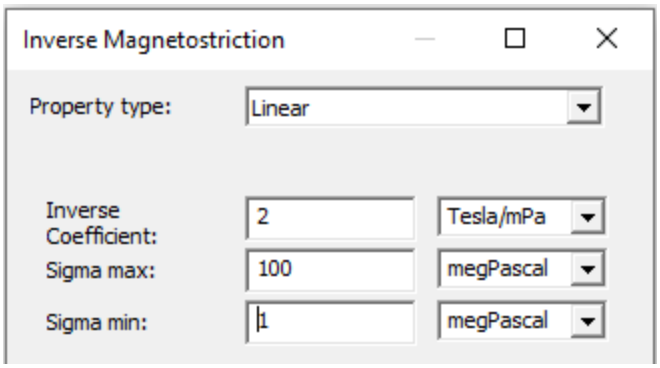
The inverse magnetostriction property supports two models: Linear, and Nonlinear. The property values you enter typically come from manufacturer data sheets.

To define these model properties, from the **View/Edit Material** window:

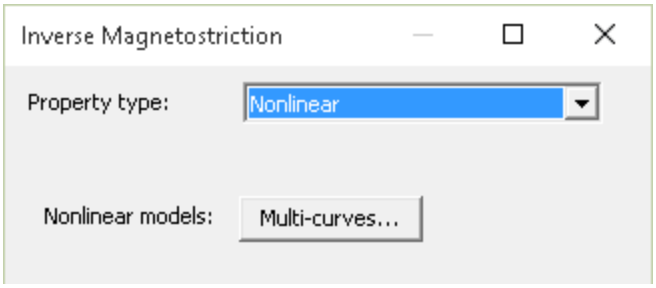
1. Click the **Edit** button to the right of **Inverse Magnetostriction Custom** to open the Inverse Magnetostriction properties dialog box.



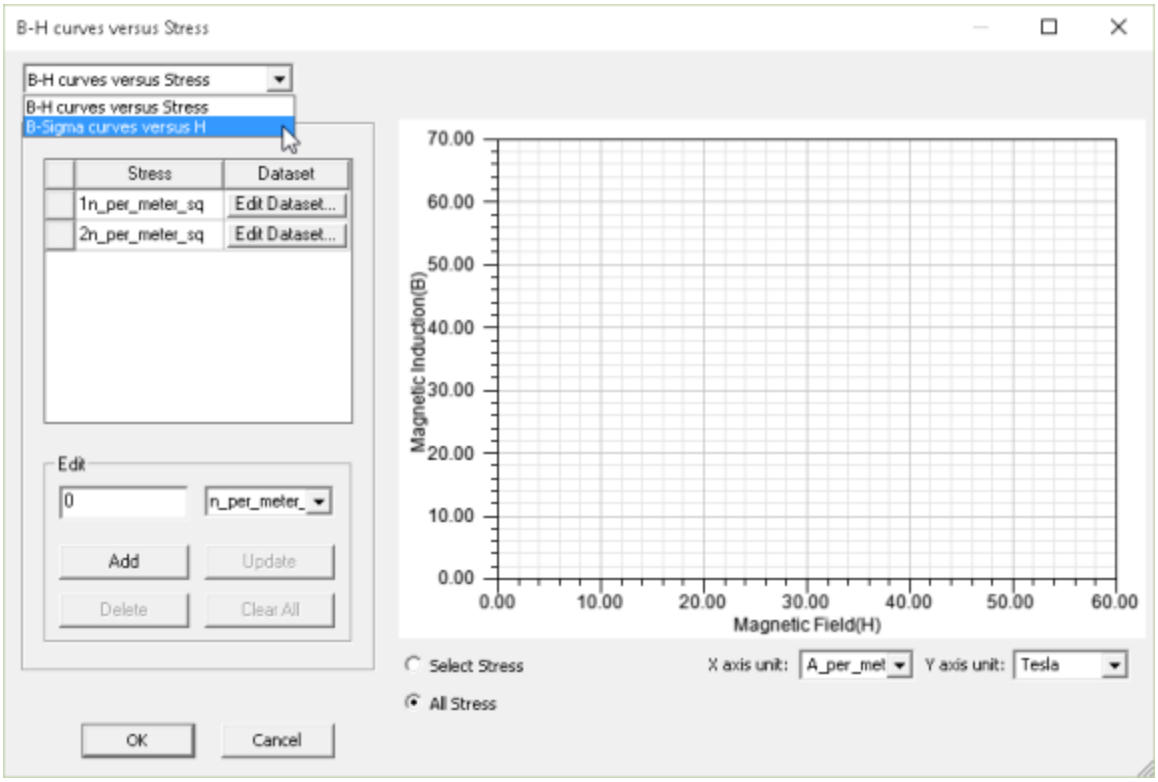
2. Choose a **Property type** from the drop-down menu: **Linear**, or **Nonlinear**, and do one of the following:
    a. For **Linear**, enter values for **Inverse Coefficient**, **Sigma max**, and **Sigma min**, where sigma represents mechanical stress.



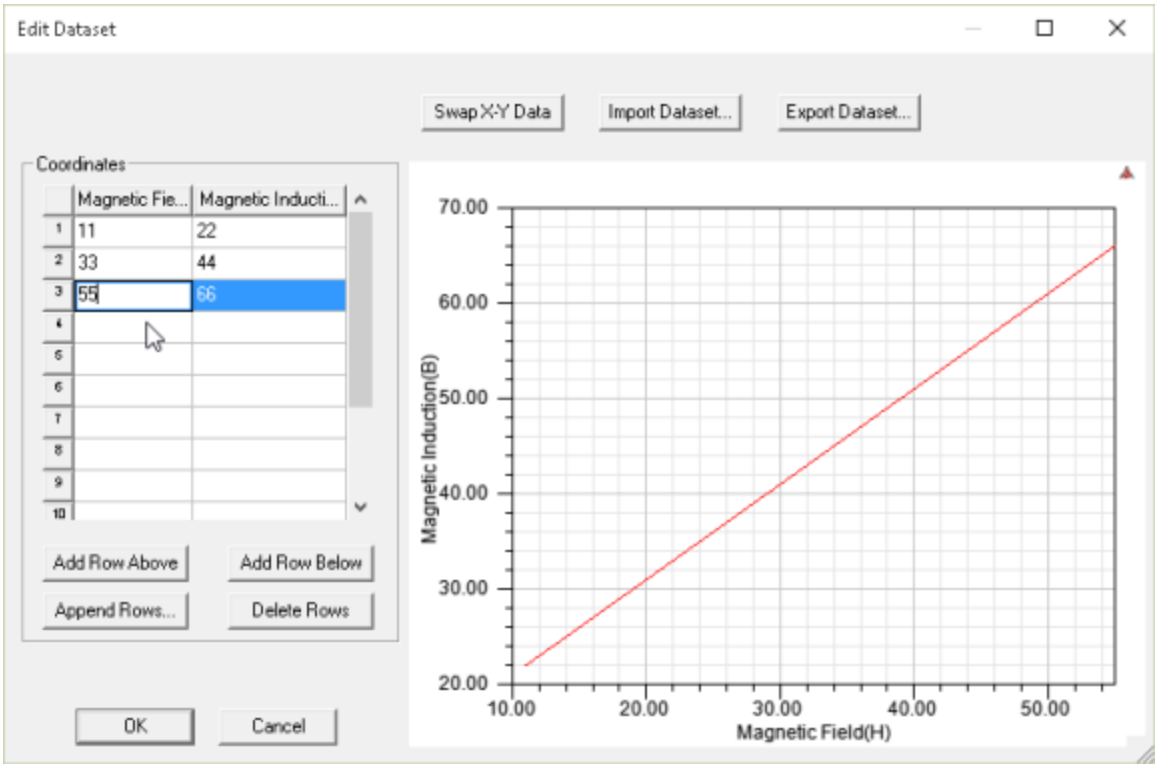
    b. For **Nonlinear**, click the **Multi-curves** button



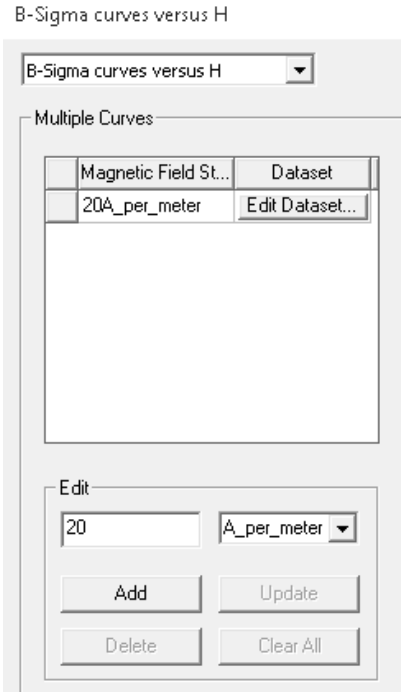
    to add one or more Stress dependent B-H curves, or H dependent B-Sigma curves.



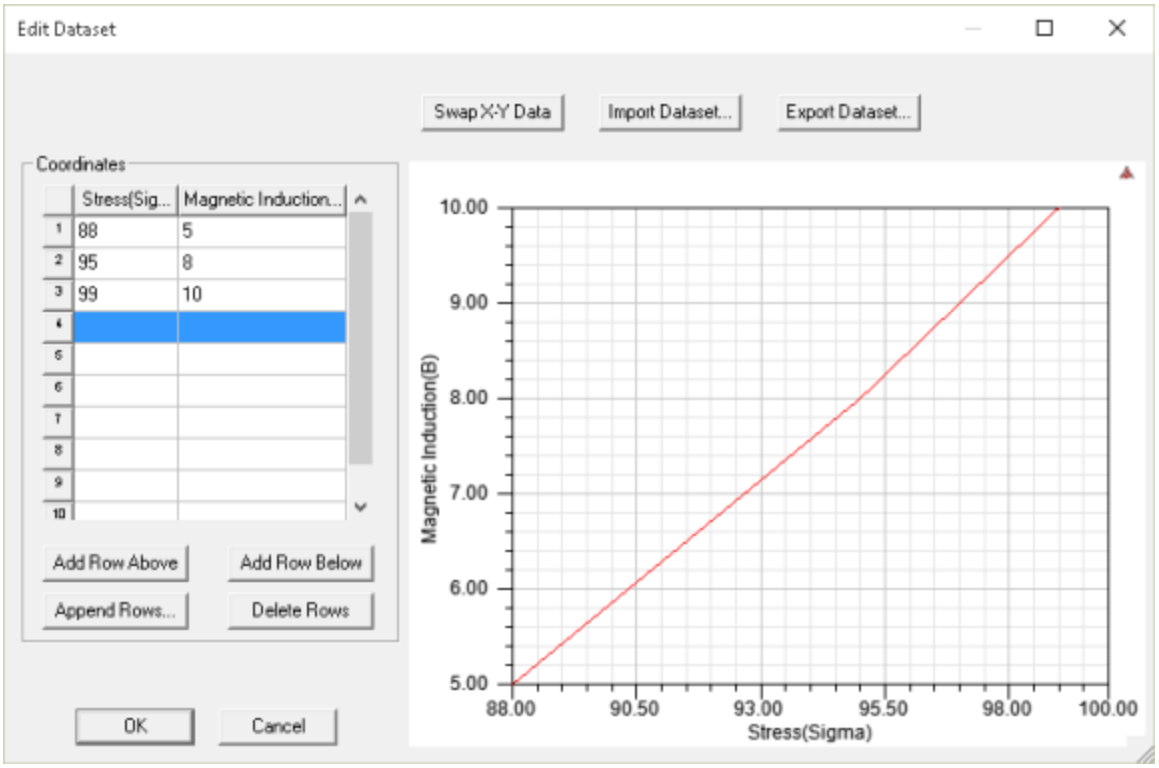
- For **B-H curves versus Stress** (as shown above) – Enter a stress value in the **Edit** text box and click **Add**; then click **Edit Dataset** to open the **Edit Dataset** dialog box, in which you enter coordinate values for the **Magnetic Field (H)** and **Magnetic Induction(B)**.



- For **B-Sigma curves versus H** (as shown below) – Enter a magnetic field strength value in the **Edit** text box and click **Add**;



then click **Edit Dataset** to open the **Edit Dataset** dialog box, in which you enter coordinate values for the **Stress (Sigma)** and **Magnetic Induction(B)**.



Note	Multiple BH curves vs stress are interpolated, but not extrapolated. Therefore, BH curves over the entire range of expected stress levels must be entered for positive stress (tensile stress pointing out of the object) and if anticipated negative stress (compressive stress pointing into the object).

When finished, click **OK** to close the dialogs and return to the **Inverse Magnetostriction** dialog.

3. Click **OK** to close the dialog and return to the **View/Edit Materials** window.

Related Topics

Magnetostriction Modeling of Magnetic Material

Magnetostriction and Inverse Magnetostriction

Calculating the Properties for Magnetostriction

Perfect Conductors

Perfect conductors are used in Maxwell to approximate the following:

- Magnetic field effects at a conductor surface when current flow is restricted to the conductor surface. This condition can exist in conductors with very high conductivity.
- The magnetic field at the surface of a conductor carrying high frequency current, modeled in the eddy current solver when the frequency is very high. Perfect conductors are used in

conjunction with Impedance Boundaries in the Eddy Current solvers to handle the following conditions:

a. The skin depth in the conductor is less than two orders of magnitude smaller than the dimensions of the structure. In models like this, the meshmaker may not be able to create a fine enough mesh in the conductor to compute eddy currents.
b. The magnetic field decays much more rapidly inside the conductor in the direction that's normal to the surface than it does in directions that are tangential to the surface.
c. The AC current source is relatively far away from the surface where eddy currents occur, compared to the size of the skin depth.

- Conductors in the Electrostatic solver where the voltage is constant in the conductor and there is no electric field penetration in the conductor.
- In the DC Conduction solver where a material contains a high conductivity.
- In conjunction with boundaries requiring the material on one side of the boundary to be the background region such as impedance or resistance boundaries.

Because of the special treatment of perfect conductors, no field solution is performed in these objects, saving time and computer resources when the approximation does not significantly change the field solution outside the conductor.

In order to define an object as a perfect conductor, do one of the following:

- Assign the Perfect Conductor material to the object in the Material Manager.
- Set the object to have a material with a conductivity greater than the conductivity defined in the *Material Threshold* section of the Design Settings dialog.

11 - Assigning Boundaries and Excitations for 3D Designs

For every project, you need to assign boundaries and excitations. For 3D designs, you can use the **Maxwell3D** menu or the project tree to do the following:

- Define boundary conditions that control how the electric or magnetic field behaves at object faces, planes of symmetry and periodicity, and edges of the problem region.
- Define excitations of voltage, charge, coil, and current.
- Identify conductors in which eddy currents are induced.
- For eddy current and magnetic transient projects, set up a winding and an external circuit connection.

Each field solver requires you to specify excitations of electric or magnetic fields and references for computing these fields.

- Magnetostatic Boundaries and Excitations
- Eddy Current Boundaries and Excitations
- Transient Boundaries and Excitations
- Transient A-Phi Formulation Boundaries and Excitations
- Electrostatic Boundaries and Excitations
- DC Conduction Boundaries and Excitations
- Electric Transient Boundaries and Excitations
- AC Conduction Boundaries and Excitations

You must specify at least one of the boundary conditions or excitations listed in these sections, so that the simulator can compute accurate values for fields and parameters.

Related Topics

Specifying the Solver Type

Selecting Objects and Faces in the Geometry

Before creating a boundary or excitation, you must specify its location by selecting the object or surface on which to assign the boundary condition or source excitation.

Defining Boundary Conditions

To assign a boundary condition:

1. Select the section of the geometry on which you want to apply the boundary condition.
2. Click **Maxwell 3D>Boundaries>Assign**, or right-click **Boundaries** in the project tree and click **Assign**.

3. Select the desired boundary type from the sub-menu. The specific boundary types you can define depend on the solver you have chosen for your project.
    - Magnetostatic Boundary Conditions
    - Electrostatic Boundary Conditions
    - DC Conduction (Electric) Boundary Conditions
    - Eddy Current Boundary Conditions
    - Transient Boundary Conditions
    - Transient A-Phi Formulation Boundary Conditions
4. Enter the name for the boundary or accept the default.
5. Depending on the boundary type, you may need to define additional parameters.
6. Click **OK** to create the new boundary.

The new boundary is added to the boundary list in the project tree.

Related Topics

Setting a Symmetry Multiplier

Defining Excitations

The specific excitations you can define depend on the solver you have chosen for your project.

- Magnetostatic Excitations
- Electrostatic (Electric) Excitations
- AC Conduction (Electric) Excitations
- DC Conduction (Electric) Excitations
- Eddy Current Excitations
- Transient Excitations
- Transient A-Phi Formulation Excitations
- Electric Transient Excitation

To assign an excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell 3D>Excitations>Assign**, or right-click **Excitations** in the project tree and click **Assign**.

   A submenu appears.
3. Select an excitation type from the submenu. The choices vary depending on the solver type you have chosen. Choices include:
    - Voltage
    - Voltage Drop
    - Charge
    - Volume Charge Density
    - Current Density
    - Current

- Permanent Magnet Field
- Sink

4. Enter the name for the excitation type or accept the default.
5. Enter the desired parameter values, units, and other settings for the excitation in the dialog box. You can also enter a mathematical function to determine values for voltage, current density, charge density, and other parameters. The Permanent Magnet Field excitation also requires you set up a link to a source project.

Note	When entering current, current density terminals, or coil terminals, the arrow associated with current in the selected object shows the direction as if a positive value is entered for the current. If a negative value is entered the actual current flow direction is opposite to what the arrow shows. Click **Swap Direction** if you want to reverse the direction of the arrow.

6. Click **OK** to create the new excitation.

The new excitation is added to the excitation list in the project tree.

Viewing Boundaries and Excitations

To view a list of all boundaries or excitations set for the project:

1. In the project tree, right-click **Boundaries** or **Excitations**.
   A shortcut menu appears.
2. Select **List** from the shortcut menu.
   The **Design List** window appears, with the tab page visible for the item you selected. This window contains tabs for **Model**, **Boundaries**, **Excitations**, **Parameters**, **Mesh Operations**, and **Analysis Setup**.
3. To view the properties for an item in the list, select the item, and click **Properties**.
4. To delete an item from the list, select the item, and click **Delete**.
5. When you are finished in the **Design List** window, click **Done** to close it.

Setting the Visibility for Boundaries and Excitations

To show or hide a boundary:

1. Click **View>Active View Visibility**.
   The **Active View Visibility** dialog box appears.
2. Click the **Boundaries** tab.
3. Select or clear the **Visibility** check box for each boundary listed.
4. Click **Done**.

To show or hide a excitation:

1. Click **View>Active View Visibility**.
   The **Active View Visibility** dialog box appears.
2. Click the **Excitations** tab.

3. Select or clear the **Visibility** check box for each excitation listed.
4. Click **Done**.

Functional Boundaries and Excitations

Functional boundaries and excitations can be used to do the following:

- Define the value of a boundary or excitation quantity (such as the voltage, magnetic field, or current density) using a mathematical relationship — such as one relating its value to that of another quantity.
- Define the value of a boundary or excitation as a function of position.
  - Current Density Excitation (Magnetostatic or Eddy Current solver type)
  - Voltage Excitation (Electric solver types)
  - Volume Charge Density Excitation (Electrostatic or Electrostatic + DC solver types)
- If the parametric analysis capability was purchased, identify which boundary or excitation quantities are to be varied during a parametric sweep. These variables are always set to constant values in the parametric analysis.

Setting Eddy Effects and Displacement Current

(Eddy Current and/or Transient Solvers)

For both eddy current and transient solvers, you may need to specify the following additional setting:

- The behavior of eddy currents and the AC magnetic field in conductors. When you activate the **Eddy Effect** setting, the solver computes the induced eddy currents.

For eddy current solvers only, you may need to specify the following additional setting:

- The **Displacement Current** on the objects in the model.

Typically, background objects are excluded from eddy and displacement current settings.

To set eddy effects:

1. Select the elements of the geometry on which you want to apply the excitation.
2. Click **Maxwell3D>Excitations>Set Eddy Effects**.
   The **Set Eddy Effect** dialog box appears.
3. For each object in the list, select or clear the check boxes for **Eddy Effect** and **Displacement Current**. When selecting multiple entries, the changes are synchronized for all selected rows.

Note	Multi-select is supported by holding the **CTRL** or **Shift** key while clicking.

4. Click the **Select Object By Name** button to enter the name of a specific object for selection.
5. Optionally, click the **Use suggested values** button to set the **Eddy Effects** for all elements recommended by Maxwell3D.
6. Click **OK**.

The **Deselect All** button will deselect all selected objects in the list.

Related Topics

Technical Notes: *Frequency Domain (Eddy Current) Solver*

Technical Notes: *3D Transient Excitations (Sources)*

Setting Core Loss for Transient and Eddy Current Solvers

For a Transient or an Eddy Current solver design, you may include core loss calculations in the simulation on any object that has a corresponding **core loss definition** (with core loss coefficient settings) in the material library.

To set core loss, in the **General** tab of the dialog box:

1. Click **Maxwell>Excitations>Set Core Loss**.

   The **Set Core Loss** dialog box appears.

   You can also open the **Set Core Loss** dialog box by right-clicking **Excitations** in the Project Manager, then selecting **Set Core Loss** on the context menu; or by right-clicking in the Modeler window and selecting **Assign Excitation> Set Core Loss**.
2. For each **Object** in the list, select or clear the check box for **Core Loss Setting** (if a corresponding core loss definition has been set). The **Defined in Material** column indicates whether the material assigned to each object contains a core loss definition. When selecting multiple entries, the changes are synchronized for all selected rows. Multi-select is supported by holding the **CTRL** or **Shift** key while clicking.
3. Click the **Select By Name** button to enter the name of a specific object for selection. Regular expressions are also supported.
4. Optionally, click the **Deselect All** button to deselect all selected objects in the list.
5. For 3D and 2D Transient solvers only, on the **Advanced** tab of the dialog box, you may select the option to **Consider core loss effect on field**. When selected, this option will estimate the core loss and subsequently modify the field solution, causing the resulting fields to be affected by the presence of the core loss. This option provides improved solution accuracy for materials exhibiting high core loss at the expense of increased solution time.

Note	• If the core loss model “hysteresis model” is chosen, then the “Consider core loss effect on field” checkbox is ignored, as the hysteresis model always models the effects of the hysteresis loss. • An iteration is required to consider loss effects on the field. For most really soft materials, the iteration is converged. However, if the loss coefficients kh and ke for magnetic steels, or Cm for power ferrites, are too large, the iteration could be diverged. In such cases, users need to double check the specified values, or not to consider core loss effects.

   The default is to exclude this effect. In this case, the field calculation is performed without the core loss and the loss calculation is performed after the field solution is complete.
6. Click **OK**.

Related Topics

Selecting Solution Quantities to Plot

Core Loss Coefficient Extraction

Core Loss Model for a Maxwell Material

Technical Notes: 3D Transient Excitations (Sources)

Technical Notes: 2D Sources/Excitations

Permanent Magnet Field Excitations for 2D and 3D Transient and Magnetostatic Solvers

(Maxwell 2D/3D Transient and Magnetostatic Solvers)

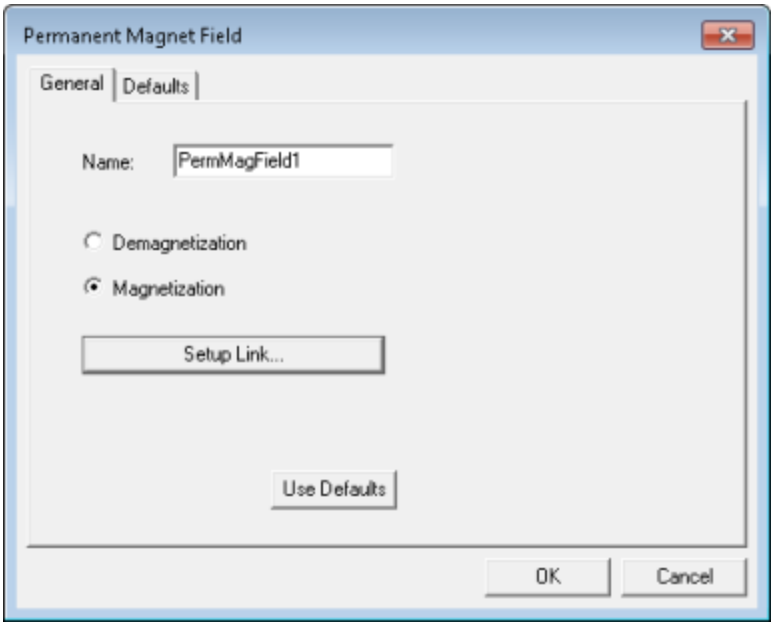
Links to demagnetization and magnetization computation *source* designs can be added as permanent magnet field excitations in *target* designs. A permanent magnet field excitation can be either *object-based* (assigned to geometry) or *design-based* (not assigned to geometry).

To add or edit a permanent magnet field excitation:

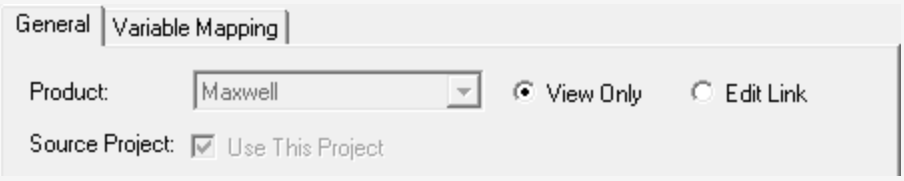
Note	The following procedure assumes that a *source* design, in which either the demagnetization or magnetization operating points computations of the magnet have been modeled, is available for linking. The *target* design object on which the Permanent Magnet Field is assigned should have a local coordinate system assigned. • This coordinate system must be right-(or left-)handed, the same as in the *source* design. • The *target* design object should be in the same quadrant of the new local coordinate system as the matching object in the *source* design.

1. Do one of the following:
   - If you are adding a new *design-based* permanent magnet field excitation, select **Maxwell 2D** or **3D>Excitations>Assign> Permanent Magnet Field**. Alternatively, in the Project Manager you can right-click **Excitations** to open its context menu, then select **Assign>Permanent Magnet Field**.
   - If you are adding a new *object-based* permanent magnet field excitation, select one or more objects (magnets) that you wish to include in the permanent magnet field excitation, then either select **Maxwell 2D** or **3D>Excitations>Assign> Permanent Magnet Field**, right-click **Excitations** in the Project tree to open its context menu and select **Assign>Permanent Magnet Field**.
   - If you are editing an existing permanent magnet field excitation, right-click on the excitation in the project tree and select **Properties** from the context menu.

2. In the **Permanent Magnet Field** dialog box, enter (or edit) the **Name** for the excitation - or accept the default.



3. Depending on how the *source* design which is to be linked has been configured using **Set Demagnetization/Magnetization Computations**, choose either **Demagnetization** or **Magnetization** for the excitation type, then click **Setup Link**.
4. By default, the **Setup Link** dialog opens on the **General** tab with only the **This project** radio button selected.

Note	If a link had previously been set up, the **Setup Link** dialog opens in **View Only** mode with all settings disabled.  To enable settings, you can select **Edit Link**.

5. To select a source project do one of the following:
   - To use the current project as the source, check **Use This Project**. This disables the **Save source path relative to:** radio buttons and the ellipsis **[...]** button and its associated text field.
   - To specify a source project file other than the current project click the ellipsis **[...]** button to open a file browser window. When you have selected the project, click the **Open** button to accept the project file for the setup. You can use the check box to

Open as read only.

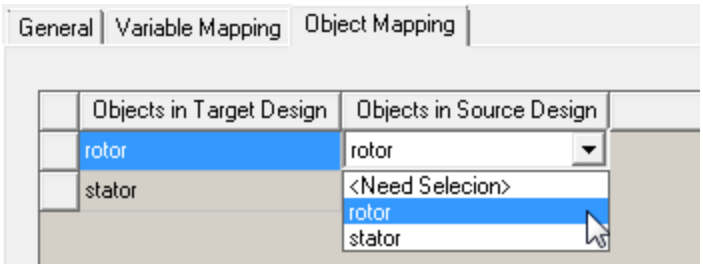
Use the radio buttons to specify whether to save the source path relative to **The project directory of selected product** or **This project**.

6. When you select a source project file, the **Source Design** and the **Source Solution** fields are filled in with default values, and their drop down menus contain any available designs and solutions. You can use the drop down menus to choose from the available designs and solutions.

   The “Default” solution is the product-dependent solution of the first Setup. That is the setup listed first in the source design's project tree (alphanumerical order). A product-specific solution of this setup becomes the default solution. In most products, it is “LastAdaptive.”
7. Use the check box to specify whether to **Simulate source design as needed**.
8. Use the check box to specify whether to **Preserve source design solution**. Note that in the extractor mode, the source project will be saved upon exit. Extractor mode means that the software is opened during the link solely for the purpose of solving.
9. The **Variable Mapping** tab lets you view any variables contained in the selected Project.

   When there are variables in the source design, you can choose to “map” these variables to constant values, expressions or variables in the target designs. Variable mapping becomes more important when the datalink type requires source and target design to be geometrically identical and source design is geometrically parameterized.

   For linked designs with variables of the same name, you can click **Map Variable by Name** to automatically map same named variables.
10. For object-based permanent magnet field excitations only, the **Object Mapping** tab lets you map objects in the target design to objects in the source design.



11. Click **OK** to close the **Setup Link** dialog box and return to the **Permanent Magnet Field** dialog.
12. Optionally, the **Defaults** tab allows you either to save the user-defined data for the excitation as the default for initializing new excitations of this type, or to clear any existing user-defined defaults and revert to standard defaults.
13. When finished, click **OK** to accept the settings and close the dialog.

Related Topics

"Permanent Magnets " on page 10-74

"Setting Demagnetization/Magnetization Computations for Source Designs " below

Setting Demagnetization/Magnetization Computations for Source Designs

(2D and 3D Magnetostatic and Transient Solvers)

When a magnetic device with permanent magnets is overloaded, or after a short circuit, irreversible demagnetization may occur due to magnetic field and temperature changes. As a result, the subsequent operating point will no longer lie along the original BH curve, but along the demagnetized recoil line at the worst operating condition.

This process can be modeled in Maxwell using **Set Demagnetization/Magnetization Computations**. The demagnetization/magnetization process of the magnet is modeled in the *source* design, and the practical normal application is modeled in the *target* design.

The linked demagnetization or magnetization designs act as sources (excitations) in the target design. These links are managed as Permanent Magnet Field excitations. Users can setup either object-based or design-based links. The Hc outputs from the source design are mapped to the target design mesh, thus eliminating the restriction of target/source designs employing the same geometries.

Note	This mapping is straightforward for non-hysteresis based models because the Hc output from the source design is a single value plus direction per mesh element. For hysteresis-based models, the Hc output from the source design is a demagnetizing curve per mesh element, and a mapping approach is non-trivial. Thus this process is not applicable when the design has any object modeled with hysteresis materials. Demagnetization/magnetization links for this type of design continues to require the same mesh between target and source designs. **Limitations** • A design can only be the target for either magnetization or demagnetization, but not both. • A design can compute either magnetized or demagnetized operating points, but not both. • Object-based and design-based links cannot coexist. • Object-based links are not supported when there is any object assigned with a hysteresis-based material model.

For the magnetization computation, Maxwell provides two possibilities: **anisotropic magnetization**, and **isotropic magnetization**.

For **anisotropic magnetization** the magnetization direction is determined by the direction specified by the user in the field for the Unit Vector input of Magnetic Coercivity on the panel of View/Edit Material during material setup.

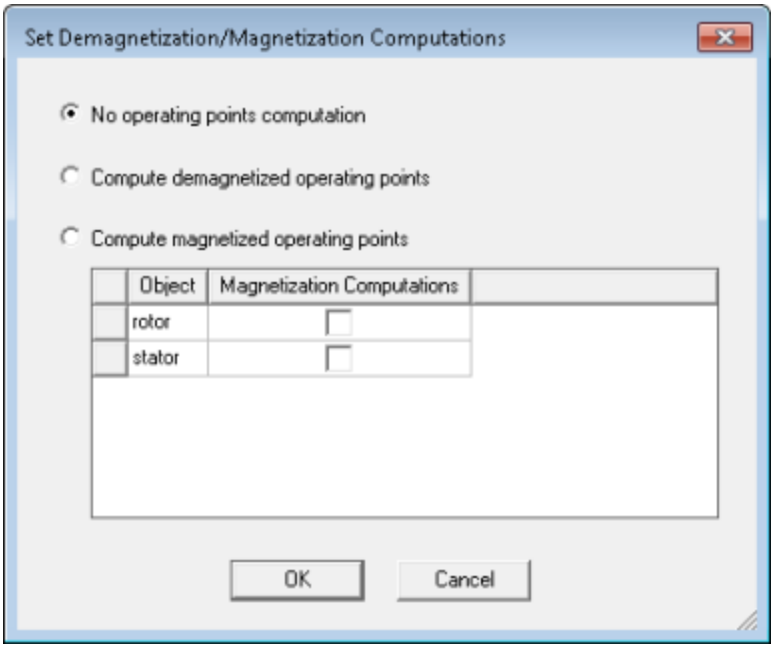
For **isotropic magnetization**, the magnetization direction is determined by the orientation of the computed magnetizing field from the field solution. In such a case, the user has to set *all three* components of the Unit Vector input of Magnetic Coercivity to *zero (0)* on the panel of View/Edit Material during material setup so that Maxwell knows the direction is computed, rather than specified by user.

General Procedure for Setting Demagnetization/Magnetization Computations for Source Designs

To set demagnetization/magnetization process computations for a *source* design:

Note	Demagnetization/Magnetization points computation settings specific to the Hysteresis Model-Based Magnetization Approach are described in the *Using the Hysteresis Model-Based Magnetization Approach* section.

1. Click **Maxwell>Excitations>Set Magnetization Computation** to open the **Set Demagnetization/Magnetization Computations** dialog box. You can also open the dialog box by right-clicking **Excitations** in the Project Manager, then selecting **Set Magnetization Computation** on the context menu; or by right-clicking in the Modeler window and selecting **Assign Excitation> Set Magnetization Computation**.



   By default, **No operating points computation** is selected.

2. Select either the **Compute demagnetized operating points** or **Compute magnetized operating points** radio button. At least one object must be selected to compute data if **Compute magnetized operating points** is selected.

Note	Multi-select is supported by holding the **CTRL** or **Shift** key while clicking.

Note	An object coordinate system (CS) must be created and assigned to each magnet object in the history tree.

3. Click **OK** to accept the settings and close the dialog.

The configured source design can now be linked to if you want to create "Permanent Magnet Field Excitations for 2D and 3D Transient and Magnetostatic Solvers " on page 11-6 .

Related Topics

Using the Hysteresis Model-Based Magnetization Approach

"Magnetostatic Solver Settings " on page 16-8

"Permanent Magnet Field Excitations for 2D and 3D Transient and Magnetostatic Solvers " on page 11-6

Technical Notes: *3D Transient Excitations (Sources)*

Technical Notes: *2D Sources/Excitations*

Using the Hysteresis Model-Based Demagnetization/Magnetization Approach

(Transient Solver Only)

> **Note:** Transient A-Phi Formulation does not support the Hysteresis Model.

Currently, demagnetization is modeled based on single demagnetization curve in the 2nd quadrant, possibly extended to the 3rd quadrant in the user-specified direction. Hysteresis-based demagnetization is able to consider magnetization and demagnetization behavior at the same time in all directions, not just user-specified initial direction.

To enable hysteresis model-based demagnetization, and set the Demagnetization/Magnetization computations for source designs:

- Define the hysteresis loop by inputting the descending branch of a hysteresis loop (not supported by defining Hysteresis Model under Core Loss Model).
- Set the initial demagnetizing direction for the starting point on the hysteresis loop by setting the direction in the field for the Unit Vector input of Magnetic Coercivity on the panel of View/Edit Material during material setup. (The initial value is inherently described by the point (Hc = 0 and B = Br).

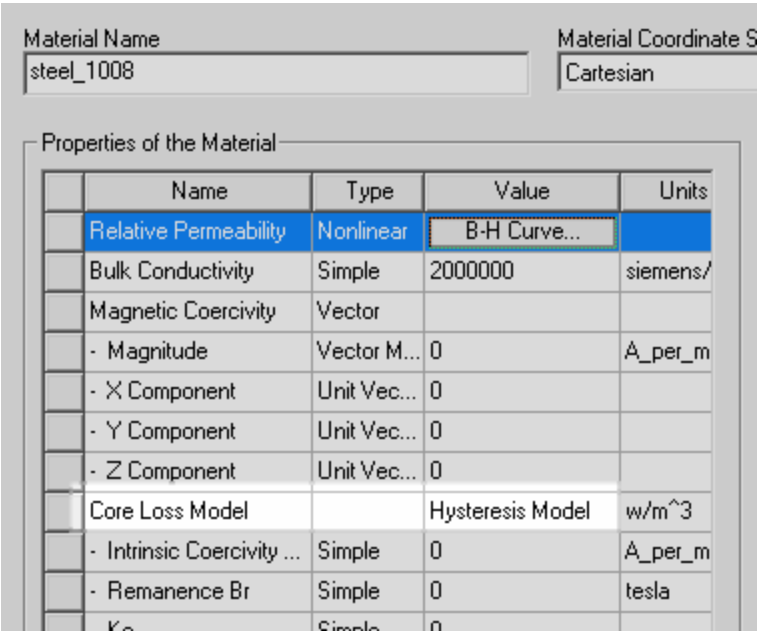
If a problem setting does not satisfy the above two conditions at the same time, the transient solver will use the “classic” approach to simulate demagnetization process if possible. Note that the valid descending branch of a hysteresis loop should be input with the same absolute values (H and B) for the 1st point and the last point. This modeling approach is not supported for TDM since the modeling of hard hysteresis material cannot be supported by Time decomposition Method (TDM).

Using the Hysteresis Model-Based Magnetization Approach

(Transient Solver Only)

> **Note:** Transient A-Phi Formulation does not support the Hysteresis Model.

With the introduction of vector hysteresis modeling capability in the transient solver, it is now possible to simulate the hysteresis model-based magnetization process if a user defines a normal nonlinear BH curve in the 1st quadrant together with the input of Hc (or additional Br for better accuracy of hysteresis loop construction) by selecting **Hysteresis Model** under **Core Loss Model** for the material.



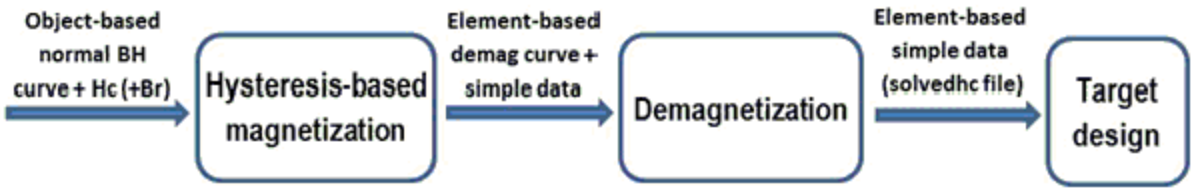
In such a case, magnetization is always considered as isotropic magnetization, that is, magnetization direction is determined by the orientation of the computed field. If a user defines only a normal nonlinear BH curve in the 1st quadrant without defining the Hysteresis model, the transient solver will use the “classic” approach to simulate magnetization process.

Classic Magnetization Approach

Using the classic magnetization approach, if the geometry in the source and target designs is the same, Maxwell maps the computed magnetization data element–by–element. However, the geometry in the source and target designs need not be identical, in which case the geometries in the two designs will be mapped by the solver using the object name. For example, magnetization data on “Box1” in the source design will be applied to the object called “Box1” in the target design. The user must ensure that the source and target object names are identical. The source design

will compute magnetization data on the selected objects, while the target design will use the dynamic data in its simulation.

Hysteresis Model-Based Magnetization Approach



Using the hysteresis model-based magnetization approach, after the completion of the last time step before stop, the hysteresis model for each element will generate the element-based demagnetization curve originated at the last operating point. As a result, the output from the magnetization modeling process is an element-based demagnetization curve, rather than a constant. Consequently an additional demagnetization process is required based on the element-based demagnetization curve. In this approach, use of the same mesh between the source and the two target designs is required due to the use of the element-based demagnetization curve as the output of magnetization simulation.

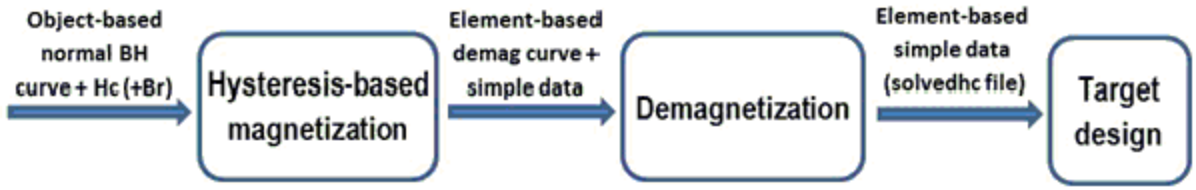
Note	• If an object is defined as hysteresis model, but not selected for magnetization, the object is no longer considered as a hysteresis material because magnetization and hysteresis modeling cannot be conducted in the same design. • Hysteresis based magnetization is always considered as isotropic magnetization, that is, its direction is determined by the orientation of the computed field. • For objects that have been selected for magnetization, but not defined as hysteresis models, the existing classic approach (*simple data* in the diagram above) will be used. Mixing hysteresis model-based magnetization method with the classic magnetization method is supported. • For the classic approach, no demagnetization step is required; while for the hysteresis model-based magnetization approach, a demagnetization step is normally required. For the demagnetization step, the input is the element-based demagnetization curve derived from the magnetization run. • When **Clean Stop** is invoked for projects in which a TDM transient source design is linked to a transient target design, after completion of the source design the solution process can continue from the target design. In such a case, the target design must be non-TDM.

Design Criteria

- The source design must have objects assigned with non-linear, non-permanent-magnet material properties.
- The same design cannot be the source for both magnetization and demagnetization links.

- The source design cannot also be a target of a demagnetization link.
- The target design cannot also be a target or source of a demagnetization link.

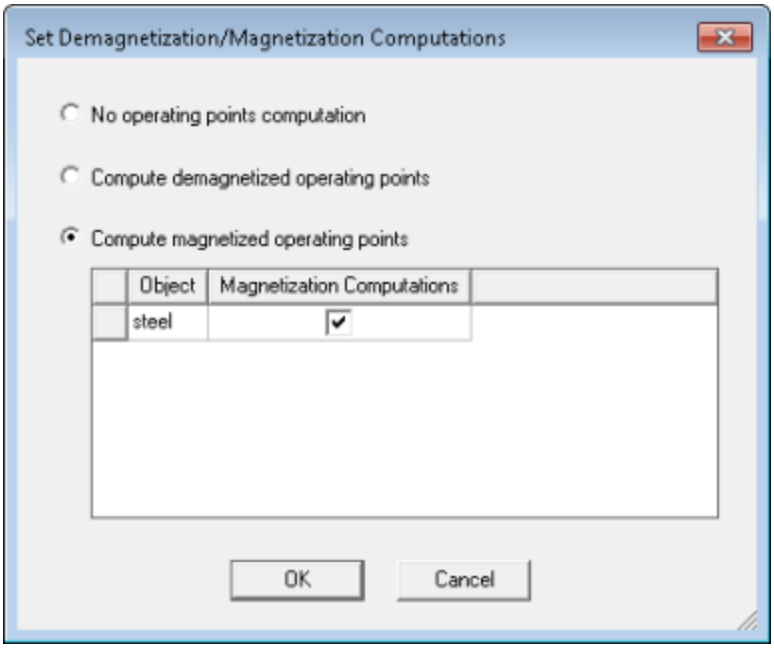
The hysteresis model-based magnetization approach requires three designs.



First Design: Set up a *Hysteresis-based magnetization computation* design (the source design)

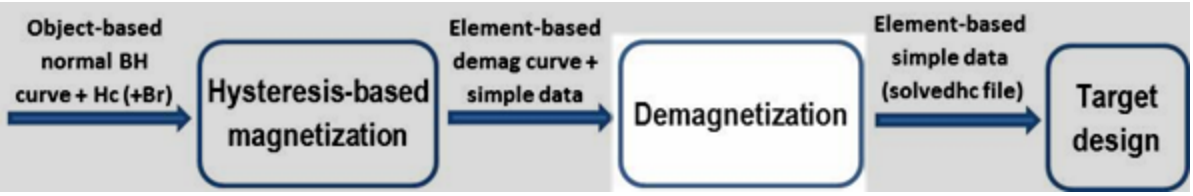
1. Create a Maxwell 2D or 3D magnetic transient design (the *Hysteresis-based magnetization* design in the figure shown above).
2. Click **Maxwell>Excitations>Set Magnetization Computation** to open the **Set Demagnetization/Magnetization Computations** dialog box .

   You can also open the **Set Demagnetization/Magnetization Computations** dialog box by right-clicking **Excitations** in the Project Manager, then selecting **Set Magnetization Computation** on the context menu; or by right-clicking in the Modeler window and selecting **Assign Excitation> Set Magnetization Computation**.
3. Select **Compute magnetized operating points**, and select the objects for magnetization and/or hysteresis-based magnetization computation, then click **OK** to accept the settings and close the dialog.

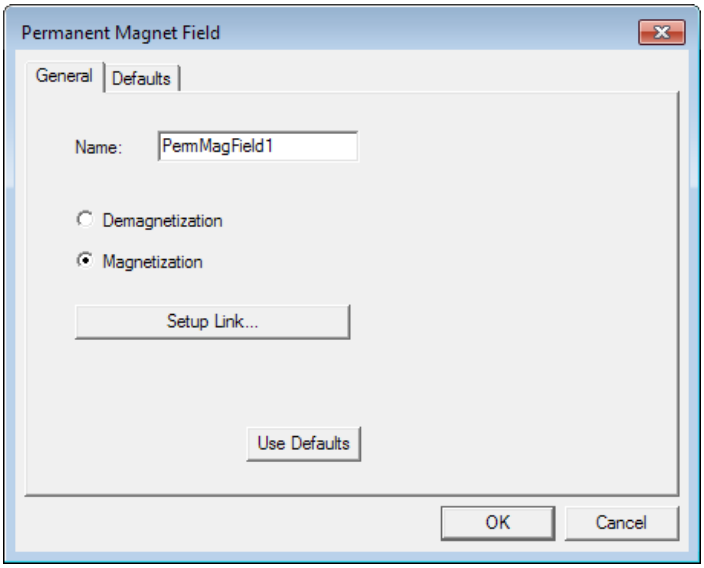


4. Solve the design and the solution will include the element-based magnetization data, and/or Hysteresis-based demagnetization curve if the design uses nonlinear materials with non-zero Hc defined.

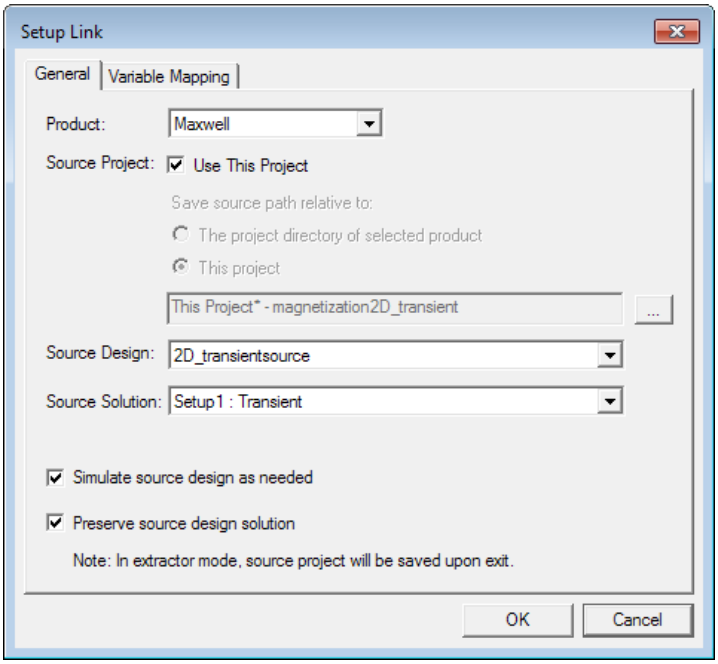
Second Design: Set up a *Demagnetization* design using the output of the Hysteresis-based magnetization computation



1. Create another Maxwell 2D or 3D magnetic transient design to be used as the *Demagnetization* design (highlighted above).
2. In the Project tree, right-click **Excitations** and select **Assign> Permanent Magnet Field**.
3. In the **Permanent Magnet Field** dialog box, select **Magnetization**, then click **Setup Link.**

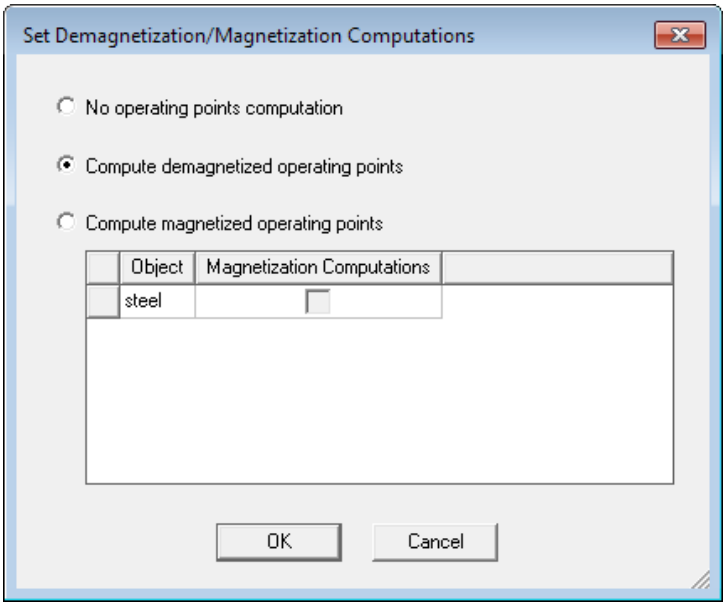


4. Set up a link to the *Hysteresis-based magnetization computation* source design (the first design you set up), which provides the hysteresis-based magnetization solution data. Check **Simulate source design as needed** and **Preserve source design solution**. If the source design is not set up to **Compute magnetized operating points**, an error message will be issued.



5. Right-click **Excitations** in the Project tree and select **Set Magnetization Computation** on the context menu to open the **Set Demagnetization/Magnetization Computations** dialog

box, and select **Compute demagnetized operating points**.

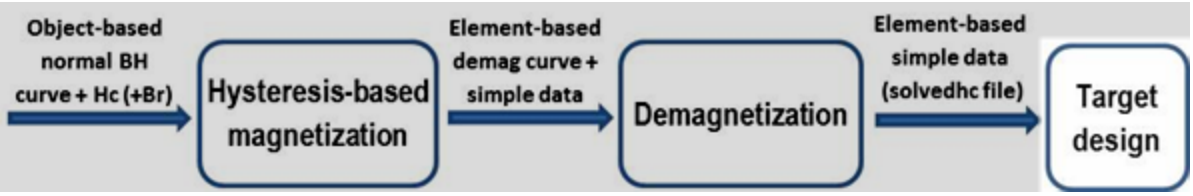


Note	Steps 6 and 7 are applicable only for *object-based* magnetization/demagnetization links. Omit these steps if *design-based* magnetization/demagnetization links are used.

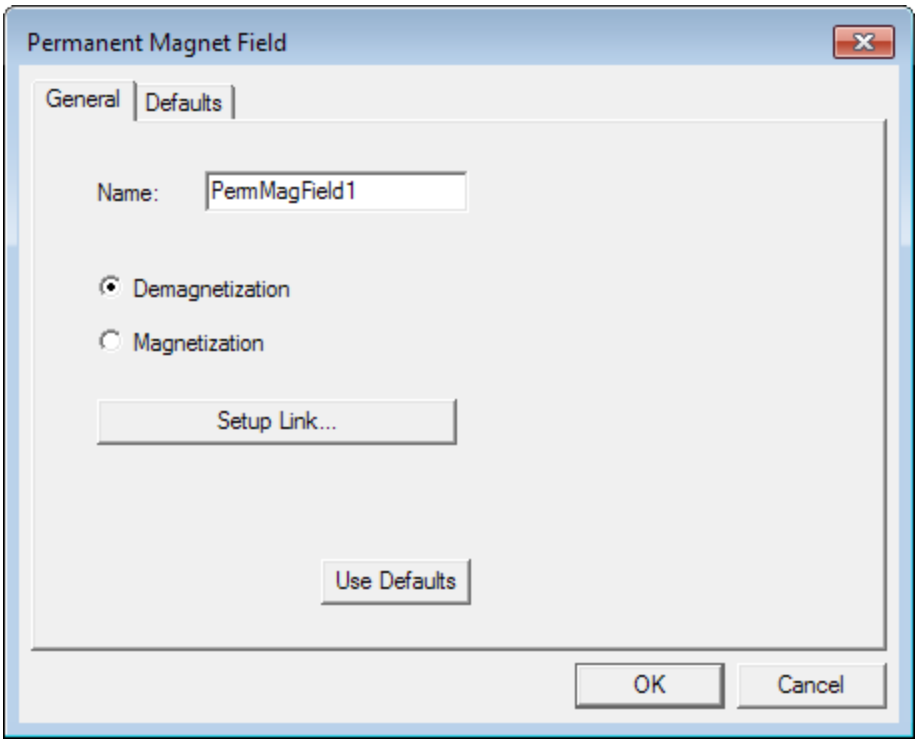
6. Open the **Solve Setup** dialog and select **Import mesh** on the **Advanced** tab to open the **Setup Link** dialog.
7. In the **Setup Link** dialog box, for the **Source Design**, choose the *Hysteresis-based magnetization computation* (the first design you set up).
8. Solve the design and the solution will include the element-based demagnetization distribution curve (which is based on the Hysteresis-based magnetization data). If the source design has not been solved, solving the current (demagnetization) design causes the source design to be solved first.

Note	To ensure that new data from the source design is used if the source design has been modified, right-click the Analysis icon of the source design in the project tree, and click **Clear Linked Data**. This flushes existing linked data so that solving the current design will force the source design to be solved first.

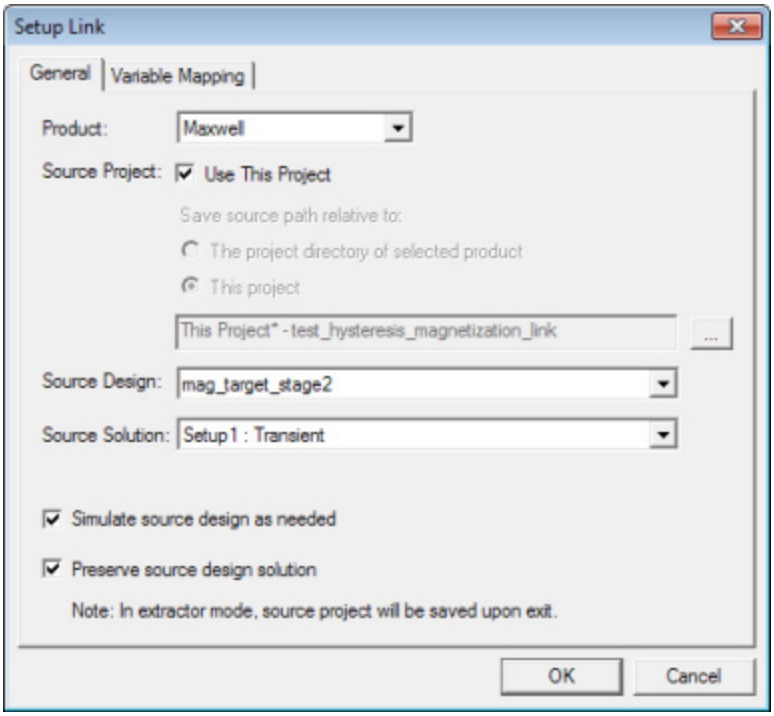
Third Design: Set up the *Target* design



1. Create a third Maxwell 2D or 3D magnetic transient design to be used at the *Target* design (highlighted above).
2. In the Project tree, right-click **Excitations** and select **Assign> Permanent Magnet Field**.
3. In the **Permanent Magnet Field** dialog box, select **Demagnetization**, then click **Setup Link.**



4. Set up a link to the *Demagnetization* design (the second design created above), which provides the dynamic demagnetization distribution solution data (based on the *Hysteresis-based magnetization* solution data from the first design). Be sure to check **Simulate source design as needed** and **Preserve source design solution**.



Note	Steps 5 and 6 are applicable only for *object-based* magnetization/demagnetization links. Omit these steps if *design-based* magnetization/demagnetization links are used.

5. Open the **Solve Setup** dialog and select **Import mesh** on the **Advanced** tab to open the **Setup Link** dialog.
6. In the **Setup Link** dialog box, for the **Source Design**, choose the *Demagnetization* design (the second design created above).
7. Solve the *Target* design. If the source designs for Hysteresis-based magnetization computation and Demagnetization are not yet solved, solving the current (target) design causes the two source designs to be solved in sequence first.

Note	To ensure that new data from the source designs is used if the source designs have been modified, right-click the Analysis icon of each source design in the project tree, and click **Clear Linked Data**. This flushes existing linked data so that solving the current design will force the source designs to be solved first.

Related Topics

Setting Demagnetization/Magnetization Computations for Source Designs

Permanent Magnet Field Excitations for 2D and 3D Transient and Magnetostatic Solvers

Showing and Verifying Conduction Paths

To show the conduction paths in the geometry:

1. Click **Maxwell>Excitations>Conduction Paths>Show Conduction Paths**.
   The **Conduction Path Visualization** window appears.
2. Click a path in the list.
   The path is highlighted in the model.
3. Click **Close** when you are done viewing conduction paths.

To verify conduction paths:

- Click **Maxwell>Excitations>Conduction Paths>Verify Conduction Paths**.
  The paths are traced in the model diagram.

Related Topics

Recalculating Conduction Paths

Recalculating Conduction Paths

Conduction paths are not recalculated unless a design changes results in a different outcome.

You can force a recalculation by using the **Maxwell>Excitations>Conduction Paths>Recalculate Conduction Paths** command.

The **Recalculate Conduction Paths** command is similar to the **Verify Conduction Paths** command. The only difference is that the **Recalculate** command recalculates every time, even if there are no changes that influence the conduction paths.

Note	The **Recalculate Conduction Paths** command does not apply for electrostatic solution types.

To recalculate conduction paths for the active project:

- Click **Maxwell>Excitations>Conduction Paths>Recalculate Conduction Paths**.

Related Topics

Showing and Verifying Conduction Paths

Modifying Boundary Conditions and Excitations

Save your project before modifying boundary conditions and excitations. You can modify boundary conditions and excitations after a solution has been generated, but you may lose some solution data.

Duplicating Boundaries and Excitations

To duplicate a boundary or excitation when its geometry is pasted or duplicated:

1. Click **Tools>Options>General Options** to open the **Options** dialog box, displaying a hierarchical list of options. Click the + next to **Maxwell 2D** and select **Boundary Assignment**.
2. Select **Duplicate boundaries/mesh operations with geometry**.
3. Click **OK**.

   All boundaries and excitations are duplicated with their associated geometries until you clear this option.

Hint	Use this option to copy and paste boundaries. For example, do the following to use the same boundary on multiple objects: 1. Select the face to which you want to assign the boundary. 2. Click **Modeler**>**Surface**>**Create Object From Face** to create a new object from the existing face. 3. Assign the boundary to the new face object. 4. Copy and paste the new face object to copy and paste the boundary.

Related Topics

Copying and Pasting Objects

Showing and Hiding Boundaries and Excitations

You can choose to show or hide a boundary or excitation's geometry, name, or vectors, in the active view window or in all view windows.

Showing and Hiding Boundaries and Excitations in Every View Window

1. Click **Maxwell>Boundaries>Visualization** or **Maxwell>Excitations>Visualization**. The **Visualization Options** window appears.
2. Select one of the following from the **Select** pull-down menu:
   - All
   - All Boundaries
   - All Excitations
   - All Ports
   - By Name
3. If you select **By Name**, enter a name in the **Select By Name** window, and click **OK**.
4. To clear the selection of all boundaries and excitations, click **Deselect All**.
5. Click **Close** to close the **Visualization Options** window.

Setting Default Values for Boundaries and Excitations

When assigning a boundary or excitation, many of the fields in the boundary and excitation dialog boxes have default values associated with them. These default values are initially set by Maxwell, but can be overridden.

To modify the default values associated with a specific boundary or excitation type:

1. Assign a boundary or excitation.
2. Modify any default values.
3. Close the boundary or excitation's dialog box.
4. Re-open the new boundary or excitation's dialog box. It now includes a **Defaults** tab.
5. Under the **Defaults** tab, click **Save Defaults**.

   The values assigned to this boundary are saved as the default values and are assigned when new boundaries of this type are created.
6. Optionally, click **Revert to Standard Defaults**.

   The default values you set for this boundary type are cleared and will revert to the default values set by Maxwell.

Reassigning Boundaries

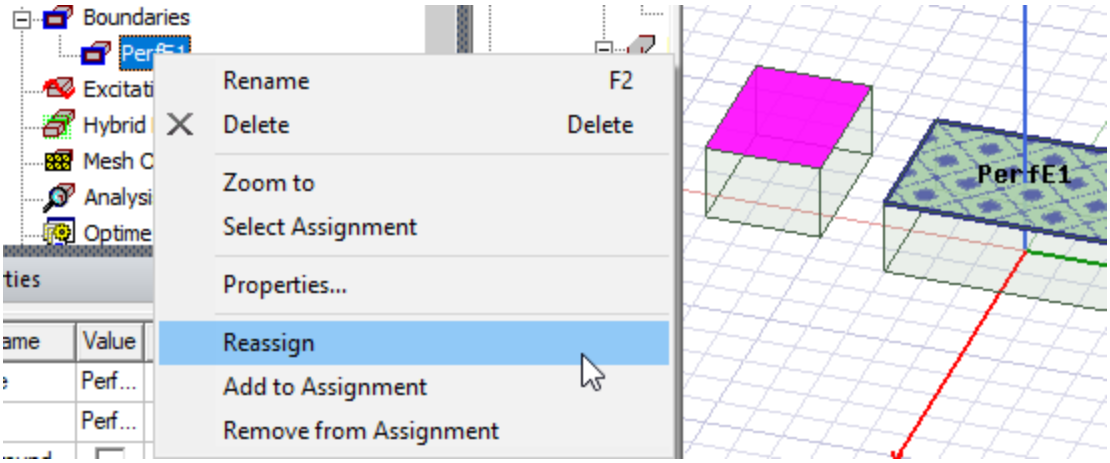
You can reassign a boundary to another surface. This is useful when you have modified objects with assigned boundaries, invalidating the boundaries. For example, if you unite two objects with assigned boundaries, the second object's boundary will become invalid because united objects maintain the characteristics of the first object selected. In this case, you would need to reassign the boundary or delete it. You can use a **Reassign Boundary** dialog box accessed through the **Maxwell>Boundary>Reassign** menu, or right-click on the Boundary of interest in the **Project** tree, and select from the shortcut menu to **Reassign**, **Add to Assignment**, or **Remove from Assignment**. You can also determine current assignments by right-clicking on the boundary on the Project tree and using the **Select Assignment** command on the short cut menu.

Prerequisites:

- You must first select the surface or object that you want to reassign, or add to an assignment, or remove from an assignment.

To use the short-cut menu to reassign, add assignment, or remove from assignment:

1. With the target surface selected, right-click on the Boundary of interest to invoke the menu.



2. Select from the menu to make the change you want.

   The boundary assignments change.

Note	When reassigning a boundary that includes vectors in its definition, Ansys Electronics Desktop attempts to preserve the vectors with the new assignment, but this is not always possible.

Related Topics

Showing and Hiding Boundaries and Excitations

Modifying Boundaries

Deleting Boundaries

Reassigning Excitations

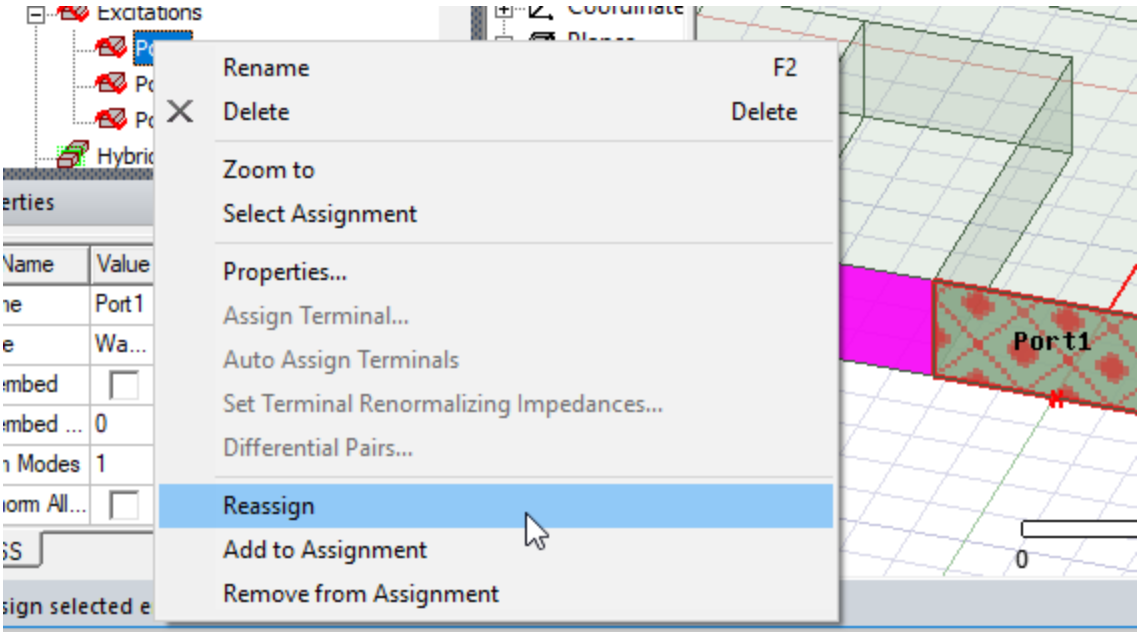
You can reassign an excitation to another surface. This is useful when you have modified objects with assigned excitations, invalidating the excitations. For example, if you unite two objects with assigned excitations, the second object's excitation will become invalid because united objects maintain the characteristics of the first object selected. In this case, you would need to reassign the excitation or delete it. You can use a **Reassign Excitation** dialog box accessed through the **Maxwell>Excitation>Reassign** menu, or right-click on the Excitation of interest in the Project tree, and select from the shortcut menu to **Reassign**, **Add to Assignment**, or **Remove from Assignment**. You can also determine a current assignment by right-clicking on the excitation in the **Project** tree and by using the **Select Assignment** command on the short cut menu.

Prerequisites:

- The surface to which you want to **Reassign**, or **Add to Assignment** must be on the same plane as existing assignment. It does not have to be in contact to **Reassign**, but must be in contact to **Add to Assignment**.
- You must first select the surface that you want to reassign or add to an assignment.

To use the short-cut menu to reassign or add an excitation:

1. With the target surface selected, right-click on the Excitation of interest to invoke the menu.



2. Select **Reassign** to change the assignment to the selected surface, or **Add to Assignment** to include surface.

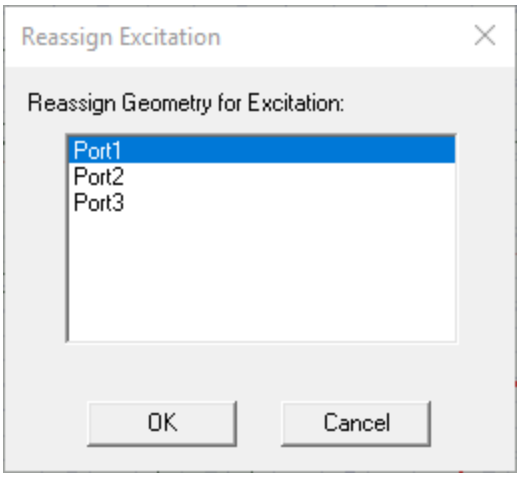
To use the shortcut menu to Remove a selected surface from an assignment:

1. With the target surface selected, right-click on the Excitation to invoke the menu.
2. Click **Remove from Assignment**.

To use the Maxwell Menu and the Reassign Excitation dialog box:

1. Select the object or object face to which you want to assign an existing excitation.
2. Click **Maxwell>Excitations>Reassign**.

   The **Reassign Excitation** window appears.



3. Select an existing excitation from the list, and then click **OK**.

   The excitation is reassigned to the object or object face.

Hint	When reassigning an excitation that includes vectors in its definition, Maxwell attempts to preserve the vectors with the new assignment, but this is not always possible.

Related Topics

Showing and Hiding Boundaries and Excitations

Modifying Boundaries

Deleting Boundaries

Deleting All Boundaries and/or Excitations

To delete all boundaries or excitations:

- Click **Maxwell>Boundaries>Delete All** or **Maxwell>Excitations>Delete All**.

Reprioritizing Boundaries

Each boundary you assign overwrites any existing boundary with which it overlaps. You can change the priority of a previously assigned boundary to be greater than a more recently assigned boundary.

The order of boundaries and excitations is important because, for any given triangle of the mesh, only one boundary or excitation can be visible to the solvers. The highest priority boundary condition applied on a certain surface overwrites the lower priority one(s) applied on the same surface. If any surfaces with boundary conditions have a partial overlap, then in the overlap region, the precedence rule applies. The highest priority boundary condition takes precedence over the lower one(s) such that for all mesh nodes on any surface carrying a boundary condition, only one boundary condition is actually enforced and visible to the solver.

Excitations cannot be reprioritized. Only boundaries can be reprioritized.

To reprioritize boundaries:

1. Click **Maxwell>Boundaries>Reprioritize**.

   The **Reprioritize Boundaries** window appears.

Note	The order the boundaries initially appear in the **Reprioritize** dialog box list indicates the order in which they were defined. The lowest priority assignment **always** appears at the top of the list.

2. Drag the boundary you want to change to the desired order of priority (to select the entity, click in the left-most column).

Note	The order of boundaries and excitations in the project tree is alphabetical. The order does not correspond to the order in which boundaries and excitations are visible to the solvers.

3. Click **OK**.

Viewing and Editing Boundary or Excitation Properties

To open the **Properties** window for a boundary or excitation that is already assigned:

1. In the project tree, under either **Boundaries** or **Excitations**, right-click a specific boundary or excitation, and select **Properties**.
   The **Properties** window for that boundary/excitation appears.
2. Make the desired changes, and click **OK**.

Set Default Boundary/Excitation Base Name

When setting boundaries and excitations on multiple selected surfaces, separate boundary or excitation definitions may be made automatically. Individual entries in the **Project Tree** will be identified with a **Base Name** and assigned a sequential numerical value appended to the base name.

The **Set Default Boundary/Excitation Base Name** dialog allows the base names for all boundary and excitation type to be specified for the current project. To set the base name for a boundary or excitation type:

1. Scroll the table to the boundary or excitation of interest.
2. Select the value in the **Default Name** column in the row of interest and enter the new base name.
3. Click **OK** to accept the changes and dismiss the dialog box.

All base names may be reset to the factory default list by clicking **Revert All**, or individuals may be reset by first selecting the **Default Name** to be reset and clicking the **Revert Selected** button.

Magnetostatic Boundaries and Excitations

Define at least one of the following as a source of static magnetic fields:

- The current in a conduction path, which can be either stranded or solid.

Note	For stranded conductors, it is assumed that the current density is uniform on the cross-section of the respective conductors. Thus, the stranded option for current excitations should mostly be used only in cases where the cross-section of conductors is constant, consistently with the assumption that the respective object is a coil built with strands of wire. For this reason, conductors (coils) could be created by either sweeping around an axis using zero segments or sweeping along a path having only true surface segments in order to create a smooth surface coil having a completely uniform cross-sectional area over its entire length.

- The current density in a conductor.
- The voltage differential across a conduction path.
- The magnetic field on an outside surface.
- A permanent magnet.

If currents or current densities are the only sources of static magnetic fields in your model, set at least one of the following as the outer boundary:

- The default boundary conditions.
- An odd symmetry (flux tangential) boundary.
- An even symmetry (flux normal) boundary.

Related Topics

Technical Notes: Magnetostatic Field Calculation

Specifying the Solver Type

Magnetostatic Boundaries

The magnetostatic field solver allows you to define the following types of boundaries:

Boundary Type	H-Field Behavior	Used to model...
Default Boundary Conditions (Natural and Neumann)	Field behaves as follows: • Natural boundaries — **H** is continuous across the boundary. • Neumann boundaries — **H** is tangential to the boundary and flux cannot cross it.	Ordinary field behavior. Initially, object interfaces are natural boundaries; outer boundaries, and excluded objects are Neumann boundaries.
Zero Tangential H Field	Flux is perpendicular if tangent components are zero or if a zero tangential H-field boundary was applied.	External magnetic fields.
Tangential H Field	The tangential components of **H** are set to pre-defined values.	External magnetic fields.
Integrated Zero Tangential H Field	For applications such as a motor with the shaft excluded, there remains a hole in the middle. In such cases, on the hole's boundary, neither the Tangential H Field boundary	Cases such as the boundary around the hole that remains when a motor's shaft is excluded.

	(where integration of the tangential H field is non-zero), nor the Zero Tangential H Field boundary (where the tangential H field is zero everywhere) can be applied because – for the hole boundary case – integration of the tangential H field is zero, but the tangential H field is not zero everywhere. Integrated Zero Tangential H Field can be used for such applications.	
Insulating	Same as Neumann, except that current cannot cross the boundary.	Thin, perfectly insulating sheets between touching conductors.
Symmetry	Field behaves as follows: • Odd Symmetry (Flux Tangential) — **H** is tangential to the boundary; its normal components are zero. • Even Symmetry (Flux Normal) — **H** is normal to the boundary; its tangential components are zero.	Planes of geometric and magnetic symmetry.
Matching (Independent and Dependent)	The H-field on the dependent boundary is forced to match the magnitude and direction (or the negative of the direction) of the H-field on the independent boundary.	Planes of symmetry in periodic structures where **H** is oblique to the boundary.

Default Boundary Conditions for a Magnetostatic Solver

These boundary conditions are automatically defined for a magnetostatic model:

- Natural boundaries are assigned to the surfaces between objects.
- Neumann boundaries are assigned to the outside edges of the problem region.

To leave a surface set to its default boundary condition, do nothing. Deleted boundary conditions and excitations automatically reset to the default boundary conditions.

Assigning a Magnetic H Field for a Magnetostatic Solver

This type of boundary defines external magnetic fields in a model. Assign it only to the outer surfaces of the problem region.

Warning	When using tangential magnetic field boundary conditions, always double-check that Ampere's law is not violated.

Note	The specific boundary types you can define depend on the solver you have chosen for your project. • Electrostatic Boundary Conditions • DC Conduction Boundary Conditions • Magnetostatic Boundary Conditions • Eddy Current Boundary Conditions

To define a zero tangential H field boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Zero Tangential H Field**.
   The **Zero Tangential H Field** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

To define a tangential H field boundary:

1. Select the section of the geometry on which you want to apply the boundary condition.
2. Click **Maxwell>Boundaries>Assign>Tangential H Field**.
   The **Tangential H Field** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the value of the field component in the x-direction in the **U (X)** box and the value of the field component in the y-direction in the **V(Y)** box.
5. In the **Coordinate System** section, do the following to define the vectors:
   a. Select **New Vector** from the **U Vector** pull-down list.
      The **Create Line** message appears asking you to draw the U vector of the coordinate system in the plane of the selected face.
   b. Click two points to specify the vector.
      The vector is set, and the **Tangential H Field** window reappears, with **Defined** listed for **U Vector**.
   c. To reverse the direction of the vector, click the **Reverse Direction** check box.

6. Optionally, click **Use Defaults** to revert to the default values in the window.
7. Click **OK** to assign the boundary to the selected object.

The new boundary is added to the boundary list in the project tree.

Assigning an Integrated Zero Tangential H Field Boundary for a Magnetostatic Solver

This type of boundary is applied when the integral of the tangential H field is zero.

Warning	When using integrated zero tangential magnetic field boundary conditions, always double-check that Ampere's law is not violated.

To define an integrated zero tangential H field boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell 3D>Boundaries>Assign>Integrated Zero Tangential H Field**.
   The **Integrated Zero Tangential H Field** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

The new boundary is added to the boundary list in the project tree.

Assigning an Insulating Boundary for a Magnetostatic Solver

This boundary condition is used to model very thin sheets of perfectly Insulating material between touching conductors. Current cannot cross an insulating boundary.

To set an insulating boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Insulating**.
   The **Insulating Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

Assigning a Symmetry Boundary Condition for a Magnetostatic Solver

This boundary condition defines a plane of geometric or magnetic symmetry in a structure. Assign it only to the outer surfaces of the problem region.

To set a symmetry boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Symmetry**.
   The **Symmetry Boundary** window appears.

3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select one of the following as the type of symmetry:

Odd (Flux Tangential)	**H** is tangential to the boundary; its normal components are zero.
Even (Flux Normal)	**H** is normal to the boundary; its tangential components are zero.

5. Optionally, click **Use Defaults** to revert to the default values in the window.
6. Click **OK** to assign the boundary to the selected object.

Warning	When using even symmetry boundaries, double-check that Ampere's law is not violated.

Assigning an Independent Boundary for a Magnetostatic Solver

Independent and dependent boundaries enable you to model planes of periodicity where the H-field at every point on the dependent boundary surface is forced to match the H-field of every corresponding point on the independent boundary surface. The transformation used to map the H-field from the independent to the dependent is determined by specifying a coordinate system on both the independent and dependent boundaries.

To set an independent boundary:

1. Select the face to which you want to assign the independent boundary.
2. Click **Maxwell>Boundaries>Assign>Matching>Independent**.
   The **Independent Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. In the **Coordinate System** section, do the following to define the coordinate system:

Note	You must specify the coordinate system in the plane on which the boundary exists. First draw the U vector of the coordinate system. Maxwell uses the U vector you draw and the normal vector of the boundary face to calculate the v-axis. Then specify the direction of the V vector.

   a. Select **New Vector** from the **U Vector** pull-down list.
      The **Create Line** message appears, asking you to draw the U vector of the coordinate system in the plane of the selected face, and the **Independent Boundary** dialog box disappears while you draw the U vector.
   b. Select the U vector's origin, which must be on the boundary's surface, in one of the following ways:
      - Click the point.
      - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.
   c. Select a point on the u-axis.
      The **Independent Boundary** dialog box reappears

5. To reverse the direction of the vector, select the **Reverse Direction** check box.
6. Click **OK**.

Maxwell computes the H-field on this boundary and maps it to the dependent boundary using the transformation defined by the independent and dependent coordinate systems.

Assigning a Dependent Boundary for a Magnetostatic Solver

Assigning a dependent boundary is the second step in creating matching boundaries. The field on the independent boundary is mapped to the dependent boundary.

Note	You must define an independent boundary before creating the dependent boundaries that are associated with it. Assign dependent boundaries only to the outer surfaces of the problem region.

To set a dependent boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Matching>Dependent**.
   The **Dependent Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select the independent boundary to which the dependent is to be assigned from the **Independent** pull-down list.
5. To specify the coordinate system, do the following in the Coordinate System section:

Note	You must specify the coordinate system in the plane on which the boundary exists. First draw the U vector of the coordinate system. Maxwell uses the U vector you draw and the normal vector of the boundary face to calculate the v-axis. Then specify the direction of the V vector.

   a. Select **New Vector** from the **U Vector** pull-down list.
      The **Create Line** message appears asking you to draw the U vector of the coordinate system in the plane of the selected face, and the **Dependent Boundary** dialog box disappears while you draw the U vector.
   b. Select the U vector's origin in one of the following ways:
      - Click the point.
      - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.

Note	The U vector's origin must be on the boundary's surface.

   c. Select a point on the u-axis in the same way.
      When the second point is identified, the vector is set, and the **Dependent Boundary** window reappears, with **Defined** listed in the **U Vector** pull-down list.
   d. To specify the direction of the V vector, select or clear the **Reverse Direction** check box.
6. In the **Relation** section, select one of the following to specify the field behavior on the boundary:

Hdep = Hind	Click this radio button if the dependent and independent boundaries have the same magnitude and direction.
Hdep = -Hind	Click this radio button if the dependent boundary field has the same magnitude as but the opposite direction from the independent boundary field.

7. Optionally, click **Use Defaults** to revert to the default values in the window.
8. Click **OK** to assign the boundary to the selected object.

Note	The origin must be a vertex point of one of the objects.

Magnetostatic Excitations

The following excitations of magnetic fields are available for magnetostatic problems:

Excitation	Type of Excitation
Voltage	The DC voltage on a surface or a sheet object.
Voltage Drop	The voltage drop across a sheet object.
Current Density	The known current density distribution in a conductor. In this case, the current density terminal must also be defined.
Current	The total current in a conductor.
Winding With Current	Current for both a stranded and solid conductor.
Winding With Resistance Voltage Drop	Resistance Voltage Drop for both a stranded and solid conductor.

In addition, permanent magnets serve as sources of magnetic fields.

Warning	If voltage sources are used as excitations, for current to flow, you must define a minimum of two voltage excitations or a voltage drop per conduction path. Each excitation must be set to a different voltage. Current flows from surfaces at higher voltages to surfaces at lower voltages.

Note	When a source conductor touches an outer boundary, Maxwell assumes current will flow perpendicular to the boundary. If you do not want current to cross the outer boundary, an insulating boundary should be used.

Assigning a Voltage Excitation for a Magnetostatic Solver

This type of excitation sets the voltage on a surface to a specific value. Use it to set up a voltage drop across a conduction path to cause current to flow.

To set a voltage excitation:

1. Select the section of the geometry on which you want to apply the excitation (typically a 2D object).
2. Click **Maxwell>Excitations>Assign>Voltage**.
   The **Voltage Excitation** window appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter a value for the voltage in the **Value** box, and select the units from the pull-down list.
5. Optionally, click **Use Defaults** to revert to the default values in the window.
6. Click **OK** to assign the excitation to the selected object.

Warning	For current to flow, you must define a minimum of two voltage excitations or a voltage drop. Each excitation must be set to a different voltage. Current flows from surfaces at higher voltages to surfaces at lower voltages.

Assigning a Voltage Drop Excitation for a Magnetostatic Solver

This type of excitation sets the voltage drop across a sheet object to a specific value. The voltage drop applies only to sheet objects:

To set a voltage drop:

1. Select the section of the geometry on which you want to apply the excitation (typically a 2D object).
2. Click **Maxwell>Excitations>Assign>Voltage Drop**.
   The **Voltage Drop Excitation** window appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the voltage drop on the surface in the **Value** box, and select the units from the pull-down list.
5. To change the direction of the voltage drop, click **Swap Direction**.
6. Optionally, click **Use Defaults** to revert to the default values in the window.
7. Click **OK** to assign the excitation to the selected object.

Assigning a Current Density Excitation for a Magnetostatic Solver

This command specifies the x-, y-, and z-components of the current density in a conduction path. If the current density is a function of position, the value is entered in ampere/m^2, even if you change the units in the problem.

To define the current density:

1. Select the section of the geometry (i.e., the conductor) on which you want to apply the excitation (typically a 3D object).
2. Click **Maxwell>Excitations>Assign>Current Density**.
   The **Current Density Excitation** window appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, specify the following options:
   a. Enter the **X Component**, **Y Component**, and **Z Component** values to define the respective components of the current density vector.
   b. Do the following in the two **Coordinate System** pull-down lists:
      - From the first pull-down list, select either **Global** or **RelativeCSx** (where x = 1,2,3… if any relative coordinate systems have been defined).
      - From the second pull-down list, select **Cartesian**, **Cylindrical**, or **Spherical**.
5. Optionally, click **Use Defaults** to revert to the default values in the window.
6. Click **OK** to assign the excitation to the selected object.

To complete the current density definition, you also need to specify any related current density terminals, which must be assigned to a 2D object or face of the respective conduction path.

To define a current density terminal:

1. Select the 2D object or face to which to assign the terminal.
2. Click **Maxwell>Excitations>Assign>Current Density Terminal**.
   The **Current Density Terminal Excitation** window appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. Click **OK** to assign the excitation to the selected object.

Assigning a Current Excitation for a Magnetostatic Solver

Specifies the total current in a conduction path. The conduction path may be contained completely within the problem region (for example, a coil), or may touch the edges of the problem region.

To set a current excitation:

1. Select the section of the geometry (i.e., the conductor) on which you want to apply the excitation (typically a face or other 2D planar object).
2. Click **Maxwell>Excitations>Assign>Current**.
   The **Current Excitation** window appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, specify the following options:
   a. Enter a value for the current in the **Value** box, and select the units from the pull-down list.
   b. Select **Solid** or **Stranded** as the **Type** of conductor. For a stranded conductor, a uniformly distributed current density is assumed.
   c. Click **Swap Direction** to change the direction of the current flow.

5. Optionally, click **Use Defaults** to revert to the default values in the window.
6. Click **OK** to assign the excitation to the selected object.

Assigning a Winding Setup for a Magnetostatic Solver

Note	• If you wish to define a winding for an Eddy Current solution type, refer to "Assigning a Winding Setup for an Eddy Current Solver " on page 11-81 . • If you wish to define a winding for a Transient solution type, refer to Assigning a Winding Setup for a Transient Solver.

You can use the predefined coil terminal(s) to define one or more current or voltage windings.

To define a winding for your model:

1. Click **Maxwell 3D>Excitations>Add Winding**. to open the **Winding** dialog box.
2. Enter a name for the winding in the **Name** box, or accept the default.
3. In the **Parameters** section:
   a. Select **Current**, or **Resistance Voltage Drop** from the **Type** drop-down menu.
   b. Select the **Solid** or **Stranded** radio button to specify the type of conductor.
   c. Enter values in the following fields (enabled according to winding type), and select the desired units.

For a current winding	For a resistance voltage drop winding[1]
Current	Resistance
	Voltage

Note	You can also type a function as an expression for any of these fields.

   d. Enter a value in the **Number of parallel branches** text box.

Note	Parallel branches for solid windings assume there are no circulating currents in the parallel branches.

4. Optionally, click **Use Defaults** to revert to the default values in the window.
5. Click **OK** to assign the excitation to the selected object.

A winding is now defined for your model.

Note	To add a coil terminal to this winding, do the following: 1. In the project tree, right-click the winding, and select **Add Terminals**. The **Add Terminals** dialog box appears, listing all coil terminals that do not already belong to that winding. 2. Select the coil terminal (s) you want to add. To select multiple terminals, press **CTRL** and click each terminal. 3. Click **OK**. To assign a coil terminal excitation and add it to this winding, do the following:

	1. Select the section of the geometry on which you want to apply the coil terminal excitation. 2. In the project tree, right-click the winding, and select **Assign Coil Terminal**. The **Coil Terminal Excitation** dialog box appears. 3. Enter a name for the excitation in the **Name** box, or accept the default. 4. In the **Parameters** section, enter the number of conductors for the coil in the **Number of Conductor**s box. **Note**: This value represents only the number of conductors inside of the selected geometry. If the Coil Terminal is cut due to symmetry, then only enter the Number of Conductors in the portion modeled. 5. Click **Swap Direction** to change the reference direction for the coil terminal. You can also change this direction in the coil terminal's Properties window by selecting either **Point into terminal**, or **Point out of terminal** for the **Direction** parameter. 6. Optionally, click **Use Defaults** to revert to the default values in the dialog box. 7. Click **OK** to assign the coil terminal excitation. The coil terminal excitation is assigned and is added to the winding. To delete all coil terminal excitations that belong to this winding, do the following: • In the project tree, right-click the winding, and select **Delete All Terminals**. All coil terminal excitations are removed from the winding and deleted from the excitations.

1. For a solid winding, the resistance term can represent: the resistance of a portion of the winding which is not modeled (for instance, end-effects), the leads connecting the winding to the source, or the source resistance. (The main winding resistance is calculated directly by the solver.) For a stranded winding, the resistance term is the complete DC resistance of the winding (since the solver does not determine resistance of a stranded winding) as well as the resistance of the end-effects, leads, source, etc. For both solid and stranded windings, the inductance term can represent: the extra inductance for a portion of the winding of the winding which is not modeled (for instance, end-effects), the leads connecting the winding to the source, or the source inductance. (The main winding inductance itself is calculated directly by the solver.)

Assigning a Coil Terminal for a Magnetostatic Solver

To assign a coil terminal as a magnetostatic excitation:

1. Select the section of the geometry on which you want to apply the excitation (typically a 2D planar object).
2. Click **Maxwell>Excitations>Assign>Coil Terminal**. (You can also right-click an existing winding in the project tree, and select **Assign Coil Terminal**.)
The **Coil Terminal Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.

4. In the **Parameters** section, enter the number of conductors for the coil terminal in the **Number of Conductors** box.

Note	This value represents only the number of conductors inside of the selected geometry. If the Coil Terminal is cut due to symmetry, then only enter the Number of Conductors in the portion modeled.

5. If you want to switch the direction for the coil terminal, click **Swap Direction**.

Note	The direction is shown in the model by a bold red arrow.

6. Optionally, click **Use Defaults** to revert to the default values in the window.
7. Click **OK** to assign the coil terminal to the selected object.

Note	To add this coil terminal to an existing winding, do the following: 1. In the project tree, right-click the coil terminal, and select **Add to Winding**. The **Add to Winding** dialog box appears. 2. Select the winding to which you ant to add the coil terminal, and click **OK**.

Note	To remove a coil terminal from an existing winding, do the following: • In the project tree, right-click the coil terminal, and select **Remove from Winding**. The terminal is removed from the winding and moved up one level in the project tree (directly beneath **Excitations**).

Related Topics

"Assigning a Winding Setup for a Magnetostatic Solver " on page 11-36

Electrostatic Boundaries and Excitations

Specify at least one of the following excitations as a source of electric fields:

- The charge on a surface or object.
- The volume charge density inside an object.
- The voltage difference between conducting surfaces and objects. Define the electric potential on each surface and object using a voltage excitation.

Include at least one of the following as a reference for computing the electric potential:

- A voltage excitation.
- An odd symmetry (flux normal) boundary.

Note	If a conductor is not assigned any excitation, it is automatically assigned a floating excitation with zero charge.

Related Topics

Technical Notes: Electric Field Calculation

Electrostatic Boundaries

The following boundary conditions are available for 3D electrostatic problems:

Boundary Type	E-Field Behavior	Used to model...
Default Boundary Conditions (Natural and Neumann)	Field behaves as follows: • Natural boundaries — The normal component of **D** changes by the amount of surface charge density. No special conditions are imposed. • Neumann boundaries — **E** is tangential to the boundary. Flux cannot cross a Neumann boundary.	Ordinary E-field behavior on boundaries. Object interfaces are initially set to natural boundaries; outer boundaries are initially set to Neumann boundaries.
Symmetry	Field behaves as follows: • Even Symmetry (Flux Tangential) — **E** is tangential to the boundary; its normal components are zero. • Odd Symmetry (Flux Normal) — **E** is normal to the boundary; its tangential components are zero.	Planes of geometric and electrical symmetry.
Matching (Independent and Dependent)	The E-field on the dependent boundary is forced to match the magnitude and direction (or the negative of the direction) of the E-field on the independent boundary.	Planes of symmetry in periodic structures where **E** is oblique to the boundary.
Insulating	E-fields can be discontinuous across the insulating boundary.	Thin layer with two input values: **relative permittivity** and **thickness** of the thin layer.

Default Boundary Conditions for an Electrostatic Solver

These boundary conditions are automatically defined for an electrostatic model:

- Natural boundaries are assigned to the surfaces between dielectrics.
- Neumann boundaries are assigned to the outside edges of the problem region.

To leave a surface set to its default boundary condition, do nothing. Deleted boundary conditions and excitations automatically revert to the default boundary conditions.

Assigning a Symmetry Boundary Condition for an Electrostatic Solver

This boundary condition defines a plane of geometric or electric symmetry in a structure. Assign it only to the outer surfaces of the problem region.

To set a symmetry boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Symmetry**.
   The **Symmetry Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select one of the following as the type of symmetry:

Odd (Flux Normal)	**E** is normal to the boundary; its tangential components are zero.
Even (Flux Tangential)	**E** is tangential to the boundary; its normal components are zero.

5. Optionally, click **Use Defaults** to revert to the default values in the window.
6. Click **OK** to assign the boundary to the selected object.

Assigning an Independent Boundary for an Electrostatic Solver

Independent and dependent boundaries enable you to model planes of periodicity where the E-field at every point on the dependent boundary surface is forced to match the E-field of every corresponding point on the independent boundary surface. The transformation used to map the E-field from the independent to the dependent is determined by specifying a coordinate system on both the independent and dependent boundaries.

To set an independent boundary:

1. Select the face to which you want to assign the independent boundary.
2. Click **Maxwell>Boundaries>Assign>Matching>Independent**.
   The **Independent Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. In the **Coordinate System** section, do the following to define the coordinate system:

Note	You must specify the coordinate system in the plane on which the boundary

	exists. First draw the U vector of the coordinate system. Maxwell uses the U vector you draw and the normal vector of the boundary face to calculate the v-axis. Then specify the direction of the V vector.

a. Select **New Vector** from the **U Vector** pull-down list.
The **Create Line** message appears, asking you to draw the U vector of the coordinate system in the plane of the selected face, and the **Independent Boundary** dialog box disappears while you draw the U vector.
b. Select the U vector's origin, which must be on the boundary's surface, in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.
c. Select a point on the u-axis.
The **Independent Boundary** dialog box reappears

5. To reverse the direction of the vector, select the **Reverse Direction** check box.
6. Click **OK**.

Maxwell computes the H-field on this boundary and maps it to the dependent boundary using the transformation defined by the independent and dependent coordinate systems.

Assigning a Dependent Boundary for an Electrostatic Solver

Assigning a dependent boundary is the second step in creating matching boundaries. The field on the Independent boundary is mapped to the dependent boundary.

Note	You must define an Independent boundary before creating the dependent boundaries that are associated with it. Assign dependent boundaries only to the outer surfaces of the problem region.

To set a dependent boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Matching>Dependent**.
The **Dependent Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select the independent boundary to which the dependent is to be assigned from the **Independent** pull-down list.
5. To specify the coordinate system, do the following in the Coordinate System section:

Note	You must specify the coordinate system in the plane on which the boundary exists. First draw the U vector of the coordinate system. Maxwell uses the U vector you draw and the normal vector of the boundary face to calculate the v-axis. Then specify the direction of the V vector.

a. Select **New Vector** from the **U Vector** pull-down list.
The **Create Line** message appears asking you to draw the U vector of the coordinate system in the plane of the selected face, and the **Dependent Boundary** dialog box

disappears while you draw the U vector.

b. Select the U vector's origin in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.

Note	The U vector's origin must be on the boundary's surface.

c. Select a point on the u-axis in the same way.
   When the second point is identified, the vector is set, and the **Dependent Boundary** window reappears, with **Defined** listed in the **U Vector** pull-down list.

d. To specify the direction of the V vector, select or clear the **Reverse Direction** check box.

6. In the **Relation** section, select one of the following to specify the field behavior on the boundary:

Edep = Eind	Click this radio button if the dependent and independent boundaries have the same magnitude and direction.
Edep = -Eind	Click this radio button if the dependent boundary field has the same magnitude as but the opposite direction from the independent boundary field.

7. Optionally, click **Use Defaults** to revert to the default values in the window.
8. Click **OK** to assign the boundary to the selected object.

Note	The origin must be a vertex point of one of the objects.

Assigning an Insulating Boundary for an Electrostatic Solver

To set an insulating boundary:

1. Select the geometry face on which you want to apply the insulating boundary condition.
2. Click **Maxwell 3D>Boundaries>Assign>Insulating**.
   The **Insulating Boundary** dialog box appears.
3. On the **Insulating Boundary** tab, enter a name for the boundary in the **Name** box, or accept the default.
4. The **Parameters** section of the **Insulating Boundary** tab allows you to set the relative permittivity and thickness of the insulating boundary. Values can be defined using variables and also support Optimetrics.
   a. Enter the desired value for **Relative Permittivity**, or accept the default value of 1.
   b. Enter the desired value for **Thickness**, or accept the default value of 0.

   Note that changing data values will invalidate the solutions.

5. Optionally, click **Use Defaults** to revert to the default values in the window.
6. Click **OK** to assign the boundary to the selected object.

Electrostatic Excitations

The following excitations are available for electrostatic problems:

Excitation	Type of Excitation
Voltage	The DC voltage on a surface or object. This excitation type is available for both electrostatic and DC conduction Electric solvers.
Charge	The total charge on a surface or object (dielectric). This excitation type is available for electrostatic solvers only.
Floating	Used to model conductors at unknown potentials. This excitation type is available for electrostatic solvers only.
Volume Charge Density	The volume charge density in an object. This excitation type is available for electrostatic solvers only.

Note	If a conductor is not assigned any excitation, it is automatically assigned a floating excitation with zero charge.

Assigning a Voltage Excitation for an Electrostatic Solver

Voltage excitations are identical to voltage boundaries in previous versions of Maxwell. Maxwell no longer has voltage boundaries in the electric solvers. Instead, a voltage excitation can be assigned to any surfaces, 2D objects, or solids. These assignments could include spatial functions. This type of excitation sets the electric potential (voltage) on a surface to a specific value.

Note	When calculating capacitance, a voltage source should be assigned to an entire object, not just a single face located on that object.

To set a voltage excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell>Excitations>Assign>Voltage**.
   The **Voltage Excitation** window appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the electric potential in the **Value** box, and select the units from the pull-down list. You can enter a numerical value or a spatial function.
5. For a spatial function, select a coordinate system from the **Coordinate System** pull-down list.

6. Optionally, click **Use Defaults** to revert to the default values in the window.
7. Click **OK** to assign the excitation to the selected object.

Assigning a Charge Excitation for an Electrostatic Solver

This type of excitation defines the total charge on a surface or object. The potential on the charge is computed during the solution.

To define a charge excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell>Excitations>Assign>Charge**.
   The **Charge Excitation** window appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the charge in the **Value** box.
5. Optionally, click **Use Defaults** to revert to the default values in the window.
6. Click **OK** to assign the excitation to the selected object.

Assigning a Floating Excitation for an Electrostatic Solver

This type of excitation models conductors at unknown potentials and specifies the total charge on the conductor.

To define a floating conductor:

1. Select the section of the geometry on which you want to apply the excitation. This can be a 2D surface or a 3D solid.
2. Click **Maxwell>Excitations>Assign>Floating**.
   The **Floating Excitation** window appears.
3. Click the **General** tab.
4. Enter a name for the excitation in the **Name** box, or accept the default.
5. In the **Parameters** section, enter the charge on the boundary in the **Value** box.
6. Optionally, click **Use Defaults** to revert to the default values in the window. You can also click the **Defaults** tab to save new default settings or to revert to the standard default values.
7. Click **OK** to assign the excitation to the selected object.

Assigning a Volume Charge Density Excitation for an Electrostatic Solver

This type of excitation defines the volume charge density on an object.

To define the charge density on a 3D object:

1. Select the 3D section of the geometry on which you want to apply the excitation.
2. Click **Maxwell>Excitations>Assign>Volume Charge Density**.
   The **Volume Charge Density Excitation** window appears.

3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the charge density in the **Value** box.
5. In the **Parameters** section, select the type of coordinate system from the **Coordinate System** pull-down list.
6. Optionally, click **Use Defaults** to revert to the default values in the window.
7. Click **OK** to assign the excitation to the selected object.

3D AC Conduction Boundaries and Excitations

The following boundaries can be specified for this solver:

- Symmetry
- (Matching) Independent/Dependent
- Thin Layer
- Insulating

The following excitations can be specified for this solver:

- Voltage
- Current
- Sink

Include at least one of the following as a reference for computing the electric potential:

- A voltage excitation (with a potential difference)
- An odd symmetry (flux normal) boundary
- A sink excitation

Related Topics

3D AC Conduction Boundaries

3D AC Conduction Excitations

Technical Notes: 3D AC Conduction Solver Theory

Specifying the Solver Type

3D AC Conduction Boundaries

The following boundary conditions are available for 3D AC conduction electric problems:

Boundary Type	E-Field Behavior	Used to model...
Default Boundary Conditions (Natural and Neumann)	Field behaves as follows: • Natural boundaries - The normal and tangent components of **J** and **E** are continuous across discontinuity surfaces. • Neumann boundaries - **E** and **J** vectors are	Object interfaces are initially set to natural boundaries; outer boundaries are initially set to

	tangential to the boundary. Flux cannot cross a Neumann boundary.	Neumann boundaries.
Symmetry	Field behaves as follows: • Even Symmetry (Flux Tangential) - **E and J vectors are** tangential to the boundary; its normal components are zero. • Odd Symmetry (Flux Normal) - **E and J vectors are** normal to the boundary; its tangential components are zero.	Planes of geometric and electrical symmetry.
Matching (Independent and Dependent)	The E-field on the dependent boundary is forced to match the magnitude of the E-field on the independent boundary if the "plus" sign is used. If the "minus" sign is used, the field on the dependent boundary will oscillate with opposite phase angle (180 degrees phase shift).	Planes of symmetry in periodic structures where **E** is oblique to the boundary.
Insulating	The current cannot cross the boundary.	Thin, perfectly insulating sheets between touching conductive domains.
Thin Layer	When two parts of the domain are separated by thin volume, it can be modeled by a Thin Layer. This boundary creates a field discontinuity (voltage drop) between the two sides of the selected surface based on the material bulk properties and thickness.	Thin and non-perfect (possibly lossy) domains.

Default Boundary Conditions for a 3D AC Conduction Solver

These boundary conditions are automatically defined for a AC conduction model:

- Natural boundaries are assigned to the surfaces between objects. The normal component of **J** and the tangential component of **E** are continuous across discontinuity surfaces.
- Neumann boundaries are assigned to the outside edges of the problem region. **E** and **J** vectors are tangential to the boundary. Flux cannot cross a Neumann boundary.

To leave a surface set to its default boundary condition, do nothing. Deleted boundary conditions and excitations automatically revert to the default boundary conditions.

Assigning a Symmetry Boundary Condition for a 3D AC Conduction Solver

This boundary condition defines a plane of geometric or electric symmetry in a structure. Assign it only to the outer surfaces of the problem region.

To set a symmetry boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell 3D>Boundaries>Assign>Symmetry**.

   The **Symmetry Boundary** dialog box appears.

3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select one of the following as the type of symmetry:

Odd (Flux Normal)	Voltages have opposite sign on the two sides of the symmetry plane.
Even (Flux Tangential)	Voltages have same sign on the two sides of the symmetry plane.

5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the boundary to the selected object.

Assigning an Independent Boundary for a 3D AC Conduction Solver

Independent and dependent boundaries enable you to model planes of periodicity where the **E**-field at every point on the dependent boundary surface is forced to match the **E**-field of every corresponding point on the independent boundary surface. The transformation used to map the E-field from the independent to the dependent is determined by specifying a coordinate system on both the independent and dependent boundaries.

To set an independent boundary:

1. Select the face to which you want to assign the independent boundary.
2. Click **Maxwell 3D>Boundaries>Assign>Matching>Independent**.
   The **Independent Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. In the **Coordinate System** section, do the following to define the coordinate system:

Note	You must specify the coordinate system in the plane on which the boundary exists. First draw the U vector of the coordinate system. Maxwell uses the U vector you draw and the normal vector of the boundary face to calculate the v-axis. Then specify the direction of the V vector.

   a. Select **New Vector** from the **U Vector** pull-down list.
   The **Create Line** message appears, asking you to draw the U vector of the coordinate system in the plane of the selected face, and the **Independent Boundary** dialog box disappears while you draw the U vector.
   b. Select the U vector's origin, which must be on the boundary's surface, in one of the following ways:
      - Click the point.
      - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.
   c. Select a point on the U-axis.
   The **Independent Boundary** dialog box reappears
5. To reverse the direction of the vector, select the **Reverse Direction** check box.
6. Click **OK**.

Maxwell computes the E-field on this boundary and maps it to the dependent boundary.

Assigning a Dependent Boundary for a 3D AC Conduction Solver

Assigning a dependent boundary is the second step in creating matching boundaries. The field on the Independent boundary is mapped to the dependent boundary.

Note	You must define an Independent boundary before creating the dependent boundaries that are associated with it. Assign dependent boundaries only to the outer surfaces of the problem region.

To set a dependent boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell 3D>Boundaries>Assign>Matching>Dependent**.

   The **Dependent Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select the independent boundary to which the dependent is to be assigned from the **Independent** pull-down list.
5. To specify the coordinate system, do the following in the Coordinate System section:

Note	You must specify the coordinate system in the plane on which the boundary exists. First draw the U vector of the coordinate system. Maxwell uses the U vector you draw and the normal vector of the boundary face to calculate the v-axis. Then specify the direction of the V vector.

   a. Select **New Vector** from the **U Vector** pull-down list.
   The **Create Line** message appears asking you to draw the U vector of the coordinate system in the plane of the selected face, and the **Dependent Boundary** dialog box disappears while you draw the U vector.

   b. Select the U vector's origin in one of the following ways:
      - Click the point.
      - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.

Note	The U vector's origin must be on the boundary's surface.

   c. Select a point on the u-axis in the same way.
   When the second point is identified, the vector is set, and the **Dependent Boundary** window reappears, with **Defined** listed in the **U Vector** pull-down list.

   d. To specify the direction of the V vector, select or clear the **Reverse Direction** check box.
6. In the **Relation** section, select one of the following to specify the field behavior on the boundary:

Edep = Eind	Click this radio button if the dependent and independent boundaries have the same

	magnitude and direction.
Edep = -Eind	Click this radio button if the dependent boundary field has the same magnitude as but the opposite direction from the independent boundary field.

7. Optionally, click **Use Defaults** to revert to the default values in the window.
8. Click **OK** to assign the boundary to the selected object.

Note	The origin must be a vertex point of one of the objects.

Assigning an Insulating Boundary for the 3D AC Conduction Solver

This boundary condition is generally used to model very thin sheets of perfectly insulating material between touching conductors, as well as infinitely thin cracks inside conductors. Current cannot cross an insulating boundary, that is, $n \cdot J = 0$.

To set an insulating boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell 3D>Boundaries>Assign>Insulating**.

   The **Insulating Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

> **Note:** If you apply the insulating boundary condition on sheet objects (inside conductors), always make sure that the respective entities are selected as faces and not as objects before attempting to assign the insulating boundary condition to them.

Assigning a Thin Layer Boundary for a 3D AC Conduction Solver

When two parts of the domain are separated by a thin volume, it can be modeled using a “thin layer” boundary condition. This boundary creates a field discontinuity (voltage drop) across the two sides of the surface based on the material bulk parameters and thickness.

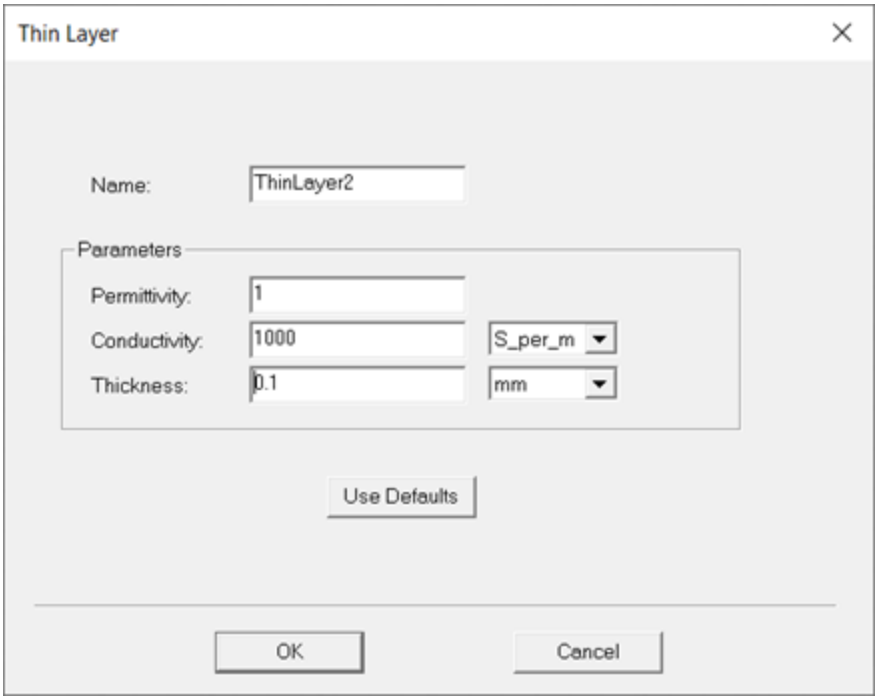
This boundary condition can be used to model thin domains, for example: the contact resistance between two conductive domains, the capacitive effect of thin insulating materials or thin lossy domains in electronic boards without needing to explicitly model (and mesh) a 3D object having a finite thickness.

> **Note:** The thin layer can be only applied to a coincident surface shared by two different bodies. It cannot be applied to a 2D cross-sectional sheet inside of a single body.

For the 3D AC conduction solver the thin layer is defined by its bulk conductivity, permittivity, and thickness.

To define a resistive sheet boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a sheet object or face of a 3D body).
2. Click **Maxwell 3D>Boundaries>Assign>Thin Layer** to open the Thin Layer dialog box.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Enter the thin layer parameters in the **Conductivity**, **Permittivity**, and **Thickness** fields, and select the units of measure. Use of variables is supported.



5. Optionally, click **Use Defaults** to revert to the default values in the window.

   The **Defaults** tab allows you to control default values. The **Save Defaults** button saves the values currently defined on the **General** tab as the defaults to be assigned to new boundaries. **Revert to Standard Defaults** clears existing user-defined values and replaces them with the standard default values.
6. Click **OK** to assign the boundary to the selected object.

The Project Manager lists the newly assigned thin layer boundary in the tree. You can select the boundary in the tree to view and edit its properties in the Properties Window. You can also double-click the boundary entry in the tree to open it for editing in the thin layer dialog box.

3D AC Conduction Excitations

The following excitations are available for 3D electric AC conduction problems:

Excitation	Type of Excitation
Voltage	The AC voltage on a surface or object.
Current	The current across a surface.
Sink	Applied only to surfaces which are grounded with reference potential value. It is the same as applying zero voltage excitation on those surfaces.

Note	All field quantities oscillate with the same frequency, equal to the frequency specified in the solution setup. The phase of quantities computed in the solution space however will be different in general.

Assigning a Voltage Excitation for a 3D AC Conduction Solver

This excitation sets the electric potential (voltage) on a surface or object to a specific value.

To set a voltage excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell 3D>Excitations>Assign>Voltage**.
   The **Voltage Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section:
   a. Enter the electric potential in the **Value** box, and select the unit from the pull-down list. You can enter a numerical value or a spatial function.

      For a spatial function, select a coordinate system from the **Coordinate System** pull-down list.
   b. Enter a value in the **Phase** text box, and select a unit from the pull-down list.
5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the excitation to the selected object.

Assigning a Current Excitation for a 3D AC Conduction Solver

To define a current across a surface:

1. Select the 2D section of the geometry on which you want to apply the excitation.
2. Click **Maxwell 3D>Excitations>Assign>Current**.
   The **Current Excitation** window appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section:
   a. Enter the current strength in the **Value** box, and select the unit from the pull-down list.
   b. Enter the phase angle in the **Phase** box, and select the unit from the pull-down list.
   c. Optionally, to change the direction of the current flow, click **Swap Direction**.

5. Optionally, click **Use Defaults** to revert to the default values in the window.
6. Click **OK** to assign the excitation to the selected object.

Assigning a Sink Excitation for a 3D AC Conduction Solver

To define a sink excitation on a surface:

1. Select the section of the geometry on which you want to apply the excitation (typically a 2D object).
2. Click **Maxwell 3D>Excitations>Assign>Sink**.
   The **Sink Excitation** window appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. Click **OK** to assign the excitation to the selected object.

DC Conduction Boundaries and Excitations

The following excitations can be specified as the source of electric fields:

- The voltage difference between surfaces and objects. Define the electric potential on each surface and object using a voltage excitation.
- Using the current excitation, define the total current flowing through a surface or a 2D object.
- Using the sink excitation, define a collection of surfaces or 2D objects as a sink. We apply sink excitations when only current excitations are defined in a conduction path and there is no voltage excitation. In fact, in this case, a sink excitation has to be defined and it ensures that the total current flowing through the outside surface of a conduction path is exactly zero.

Include at least one of the following as a reference for computing the electric potential:

- A voltage excitation.
- A sink excitation.
- An odd symmetry (flux normal) boundary.

Related Topics

Technical Notes: Electric Field Calculation

Specifying the Solver Type

DC Conduction Boundaries

The following boundary conditions are available for DC conduction electric problems:

Boundary Type	E-Field Behavior	Used to model...
Default Boundary	Field behaves as follows: • Natural boundaries — The normal	Ordinary E-field behavior on boundaries. Object

Conditions (Natural and Neumann)	component of **D** changes by the amount of surface charge density. No special conditions are imposed. • Neumann boundaries — **E** is tangential to the boundary. Flux cannot cross a Neumann boundary.	interfaces are initially set to natural boundaries; outer boundaries are initially set to Neumann boundaries.
Insulating	Same as Neumann, except that current cannot cross the boundary. An insulating boundary is *only* available for electrostatic solutions that include a DC conduction analysis.	Thin, perfectly insulating sheets between touching conductors.
Symmetry	Field behaves as follows: • Even Symmetry (Flux Tangential) — **E** is tangential to the boundary; its normal components are zero. • Odd Symmetry (Flux Normal) — **E** is normal to the boundary; its tangential components are zero.	Planes of geometric and electrical symmetry.
Matching (Independent and Dependent)	The E-field on the dependent boundary is forced to match the magnitude and direction (or the negative of the direction) of the E-field on the independent boundary.	Planes of symmetry in periodic structures where **E** is oblique to the boundary.
Thin Layer	When two parts of the domain are separated by a thin volume, it can be modeled by a Thin Layer. This boundary creates a field discontinuity (voltage drop) between the two sides of the selected surface based on the material bulk properties and thickness.	Thin conducting domains

Default Boundary Conditions for a DC Conduction Solver

These boundary conditions are automatically defined for a DC conduction model:

- Natural boundaries are assigned to the surfaces between objects.
- Neumann boundaries are assigned to the outside edges of the problem region.

To leave a surface set to its default boundary condition, do nothing. Deleted boundary conditions and excitations automatically revert to the default boundary conditions.

Assigning an Insulating Boundary for a DC Conduction Solver

This boundary condition is used to model very thin sheets of perfectly insulating material between touching conductors. Current cannot cross an insulating boundary.

An insulating boundary is *only* available for electrostatic solutions that include a DC conduction analysis.

To set an insulating boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Insulating**.
   The **Insulating Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

Assigning a Symmetry Boundary Condition for a DC Conduction Solver

This boundary condition defines a plane of geometric or electric symmetry in a structure. Assign it only to the outer surfaces of the problem region.

To set a symmetry boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Symmetry**.
   The **Symmetry Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select one of the following as the type of symmetry:

Odd (Flux Normal)	Current flows in opposite directions on either side of the symmetry plane.
Even (Flux Tangential)	Current flows in the same direction on both sides of the symmetry plane.

5. Optionally, click **Use Defaults** to revert to the default values in the window.
6. Click **OK** to assign the boundary to the selected object.

Warning	When using even symmetry boundaries, double-check the following two points: • Do not violate Ampere's law! • All magnetic field boundaries must be connected to each other. Defining disconnected magnetic field boundaries and even symmetry boundaries can produce unexpected results, as there is no unique solution to such problems.

Assigning a Independent Boundary for a DC Conduction Solver

Independent and dependent boundaries enable you to model planes of periodicity where the H-field at every point on the dependent boundary surface is forced to match the H-field of every corresponding point on the independent boundary surface. The transformation used to map the H-field from the independent to the dependent is determined by specifying a coordinate system on both the independent and dependent boundaries.

To set an independent boundary:

1. Select the face to which you want to assign the independent boundary.
2. Click **Maxwell>Boundaries>Assign>Matching>Independent**.
   The **Independent Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. In the **Coordinate System** section, do the following to define the coordinate system:

Note	You must specify the coordinate system in the plane on which the boundary exists. First draw the U vector of the coordinate system. Maxwell uses the U vector you draw and the normal vector of the boundary face to calculate the v-axis. Then specify the direction of the V vector.

   a. Select **New Vector** from the **U Vector** pull-down list.
      The **Create Line** message appears, asking you to draw the U vector of the coordinate system in the plane of the selected face, and the **Independent Boundary** dialog box disappears while you draw the U vector.
   b. Select the U vector's origin, which must be on the boundary's surface, in one of the following ways:
      - Click the point.
      - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.
   c. Select a point on the u-axis.
      The **Independent Boundary** dialog box reappears
5. To reverse the direction of the vector, select the **Reverse Direction** check box.
6. Click **OK**.

Maxwell computes the E-field on this boundary and maps it to the dependent boundary using the transformation defined by the independent and dependent coordinate systems.

Assigning a Dependent Boundary for a DC Conduction Solver

Assigning a dependent boundary is the second step in creating matching boundaries. The field on the Independent boundary is mapped to the dependent boundary.

Note	You must define a Independent boundary before creating the dependent boundaries that are associated with it. Assign dependent boundaries only to the outer surfaces of the problem region.

To set a dependent boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Matching>Dependent**.
   The **Dependent Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select the independent boundary to which the dependent is to be assigned from the **Independent** pull-down list.
5. To specify the coordinate system, do the following in the Coordinate System section:

Note	You must specify the coordinate system in the plane on which the boundary exists. First draw the U vector of the coordinate system. Maxwell uses the U vector you draw and the normal vector of the boundary face to calculate the v-axis. Then specify the direction of the V vector.

   a. Select **New Vector** from the **U Vector** pull-down list.
      The **Create Line** message appears asking you to draw the U vector of the coordinate system in the plane of the selected face, and the **Dependent Boundary** dialog box disappears while you draw the U vector.
   b. Select the U vector's origin in one of the following ways:
      - Click the point.
      - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.

Note	The U vector's origin must be on the boundary's surface.

   c. Select a point on the u-axis in the same way.
      When the second point is identified, the vector is set, and the **Dependent Boundary** window reappears, with **Defined** listed in the **U Vector** pull-down list.
   d. To specify the direction of the V vector, select or clear the **Reverse Direction** check box.

Note	When entering current, current density terminals, or coil terminals, the arrow associated with current in the selected object shows the direction as if a positive value is entered for the current. If a negative value is entered the actual current flow direction is opposite to what the arrow shows. Click **Reverse Direction** if you want to reverse the direction of the arrow.

6. In the **Relation** section, select one of the following to specify the field behavior on the boundary:

Edep = Eind	Click this radio button if the dependent and independent boundaries have the same magnitude and direction.
Edep = -Eind	Click this radio button if the dependent boundary field has the same magnitude as but the opposite direction from the independent boundary field.

7. Optionally, click **Use Defaults** to revert to the default values in the window.
8. Click **OK** to assign the boundary to the selected object.

Note	The origin must be a vertex point of one of the objects.

Assigning a Thin Layer Boundary for a 3D DC Conduction Solver

When two parts of the domain are separated by a thin volume, it can be modeled using a “thin layer” boundary condition. This boundary creates a field discontinuity (voltage drop) across the two sides of the surface based on the material bulk parameters and thickness.

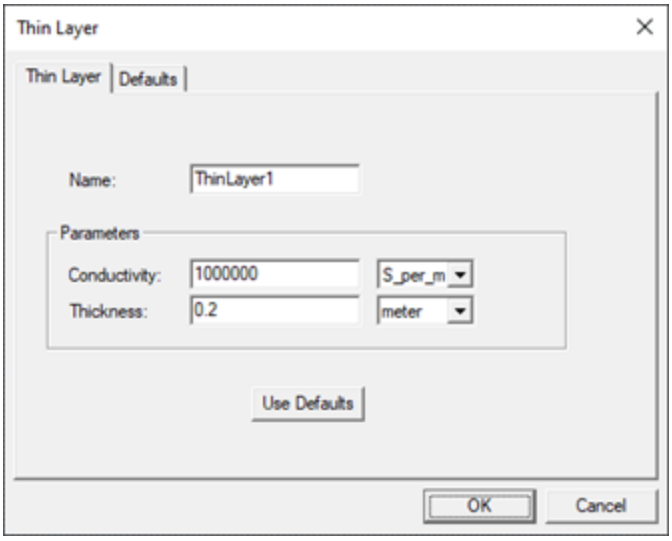
This boundary condition can be used to model thin domains, for example, the contact resistance between two conductive domains without needing to explicitly model (and mesh) a 3D object having a finite thickness.

> **Note:** The thin layer can be only applied to a coincident surface shared by two different bodies. It cannot be applied to a 2D cross-sectional sheet inside of a single body.

For the 3D DC conduction solver, the thin layer is defined by its bulk conductivity, and thickness.

To define a thin layer boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a sheet object or face of a 3D body).
2. Click **Maxwell 3D>Boundaries>Assign>Thin Layer** to open the Thin Layer dialog box.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Enter the thin layer parameters in the **Conductivity** and **Thickness** fields, and select the units of measure. Use of variables is supported.



5. Optionally, click **Use Defaults** to revert to the default values in the window.

   The **Defaults** tab allows you to control default values. The **Save Defaults** button saves the values currently defined on the **Thin Layer** tab as the defaults to be assigned to new boundaries. **Revert to Standard Defaults** clears existing user-defined values and replaces them with the standard default values.
6. Click **OK** to assign the boundary to the selected object.

The Project Manager lists the newly assigned thin layer boundary in the tree. You can select the boundary in the tree to view and edit its properties in the Properties Window. You can also double-click the boundary entry in the tree to open it for editing in the thin layer dialog box.

DC Conduction Excitations

The following excitations are available for electric DC conduction problems:

Excitation	Type of Excitation
Voltage	The DC voltage on a surface or object. This excitation type is available for both electrostatic and DC conduction solvers.
Current	The current across a 2D object. This excitation type is available for DC conduction solvers only. A current excitation must be applied across a 2D surface.
Sink	This excitation type is available for DC conduction solvers only.

For a DC conduction solution, if a current excitation is assigned, at least one sink or voltage excitation must also be defined. However, you cannot define both a voltage excitation and a sink excitation at the same time.

Assigning a Voltage Excitation for a DC Conduction Solver

This type of excitation sets the electric potential (voltage) on a surface or object to a specific value.

To set a voltage excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell>Excitations>Assign>Voltage**.
   The **Voltage Excitation** window appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the electric potential in the **Value** box, and select the units from the pull-down list. You can enter a numerical value or a spatial function.
5. For a spatial function, select a coordinate system from the **Coordinate System** pull-down list.
6. Optionally, click **Use Defaults** to revert to the default values in the window.
7. Click **OK** to assign the excitation to the selected object.

Assigning a Current Excitation for a DC Conduction Solver

To define a current across a surface:

1. Select the 2D section of the geometry on which you want to apply the excitation.
2. Click **Maxwell>Excitations>Assign>Current**.
   The **Current Excitation** window appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the current strength in the **Value** box.
5. Optionally, to change the direction of the current flow, click **Swap Direction**.
6. Optionally, click **Use Defaults** to revert to the default values in the window.
7. Click **OK** to assign the excitation to the selected object.

Assigning a Sink Excitation for a DC Conduction Solver

To define a sink excitation on a surface:

1. Select the section of the geometry on which you want to apply the excitation (typically a 2D object).
2. Click **Maxwell>Excitations>Assign>Sink**.
   The **Sink Excitation** window appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. Click **OK** to assign the excitation to the selected object.

DC Conduction + Electrostatic Boundaries and Excitations

In the case of a problem when you are interested in the electric solution in both the conductors and the insulators, you can define a **DC Conduction + Electrostatic**solution type. The Insulator/Conductor threshold defines which objects are conductors or insulators. The solver first solves the conduction problem in all conduction paths and then, using the calculated voltage on the conductors as a voltage excitation, computes the electrostatic field in the insulators. As a consequence, you can define the DC-conduction excitations and boundary conditions in the conducting regions and the electrostatic excitations and boundary conditions in the insulators.

Related Topics

Technical Notes: Electric Field Calculation

Specifying the Solver Type

Eddy Current Boundaries and Excitations

Specify at least one of the following as a source of AC magnetic fields in your model:

- The current in a conduction path, which can be either stranded or solid.

Note	For stranded conductors, it is assumed that the current density is uniform on the cross-section of the respective conductors. Thus, the stranded option for current excitations should mostly be used only in cases where the cross-section of conductors is constant, consistently with the assumption that the respective object is a coil built with strands of wire. For this reason, conductors (coils) could be created by either sweeping around an axis using zero segments or sweeping along a path having only true surface segments in order to create a smooth surface coil having a completely uniform cross-sectional area over its entire length.

- The current density in a conductor.
- The magnetic field on an outside surface.

If currents or current densities are the only sources of AC magnetic fields in your model, set at least one outer boundary to the following:

- The default boundary conditions.
- An odd symmetry (flux tangential) boundary.
- An even symmetry (flux normal) boundary.

Related Topics

3D Eddy Current Boundaries

3D Eddy Current Excitations

Technical Notes: Frequency Domain (Eddy Current) Solver

Specifying the Solver Type

Eddy Current Boundaries

The eddy current field solver allows you to define the following types of boundaries:

Boundary Type	H-Field Behavior	Used to model…
Default Boundary Conditions (Natural and Neumann)	Field behaves as follows: • Natural boundaries — **H** is continuous across the boundary. • Neumann boundaries — **H** is tangential to the boundary and flux cannot cross it.	Ordinary field behavior. Initially, object interfaces are natural boundaries; outer boundaries and excluded objects are Neumann

Boundary Type	H-Field Behavior	Used to model…
		boundaries.
Zero Tangential H Field	Flux is perpendicular if tangent components are zero or if a zero tangential H-field boundary condition was applied.	External AC magnetic fields.
Tangential H Field	The tangential components of **H** are set to pre-defined values.	External AC magnetic fields
Integrated Zero Tangential H Field	For applications such as a motor with the shaft excluded, there remains a hole in the middle. In such cases, on the hole's boundary, neither the Tangential H Field boundary (where integration of the tangential H field is non-zero), nor the Zero Tangential H Field boundary (where the tangential H field is zero everywhere) can be applied because – for the hole boundary case – integration of the tangential H field is zero, but the tangential H field is not zero everywhere. Integrated Zero Tangential H Field can be used for such applications.	Cases such as the boundary around the hole left when a motor's shaft is excluded.
Insulating	Same as Neumann, except that current cannot cross the boundary.	Perfectly insulating sheets between conductors.
Symmetry	Field behaves as follows: • Odd Symmetry (Flux Tangential) — **H** is tangential to the boundary; its normal components are zero. • Even Symmetry (Flux Normal) — **H** is normal to the boundary; its tangential components are zero.	Planes of geometric and magnetic symmetry.
Matching (Independent and Dependent)	The H-field on the dependent boundary is forced to match the magnitude and direction (or the negative of the direction) of the H-field on the independent boundary.	Planes of symmetry in periodic structures where **H** is oblique to the boundary.
Radiation	No restrictions on the field behavior.	Unbounded eddy currents.
Impedance	Includes the effect of induced currents beyond the boundary surface.	Conductors with very small skin

Boundary Type	H-Field Behavior	Used to model…
		depths.
Resistive Sheet	When two conductors are in contact within a conduction path, a voltage drop normally exists across the contact surface due to the imperfection of the contact between the two conductors. This boundary condition assigns a resistive sheet to consider the impact of this kind of voltage drop. The resistive sheet is defined by a lumped resistance in ohms.	Voltage drop and loss on the contact surface between two different objects within a conduction path.

Default Boundary Conditions for the Eddy Current Solver

These boundary conditions are automatically defined for an eddy current model:

- Natural boundaries are assigned to the surfaces between objects.
- Neumann boundaries are assigned to the outside edges of the problem region.

To leave a surface set to its default boundary condition, do nothing. Deleted boundary conditions and excitations automatically reset to the default boundary conditions.

Assigning a Magnetic H Field for the Eddy Current Solver

This type of boundary defines external magnetic fields in a model. Assign it only to the outer surfaces of the problem region. Regardless of the model's drawing units, magnetic field values are entered in s/m.

Warning	When using tangential magnetic field boundary conditions, always double-check that Ampere's law is not violated!

To define a zero tangential H field boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Zero Tangential H Field**.
   The **Zero Tangential H Field** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

To define a tangential H field boundary:

1. Select the section of the geometry on which you want to apply the boundary condition.
2. Click **Maxwell>Boundaries>Assign>Tangential H Field**.
   The **Tangential H Field** window appears.

3. Enter a name for the boundary in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the value of the field component in the x-direction in the **U (X)** box and the value of the field component in the y-direction in the **V(Y)** box.
5. In the **Coordinate System** section, do the following define the vectors:
   a. Select **New Vector** from the **U Vector** pull-down list.
   The **Create Line** message appears asking you to draw the U vector of the coordinate system in the plane of the selected face.
   b. Click two points to specify the vector.
   The vector is set, and the **Tangential H Field** window reappears, with **Defined** listed for **U Vector**.
   c. To reverse the direction of the vector, click the **Reverse Direction** check box.
6. Optionally, click **Use Defaults** to revert to the default values in the window.
7. Click **OK** to assign the boundary to the selected object.

The new boundary is added to the boundary list in the project tree.

Assigning an Integrated Zero Tangential H Field Boundary for the Eddy Current Solver

This type of boundary is applied when the integral of the tangential H field is zero.

Warning	When using integrated zero tangential magnetic field boundary conditions, always double-check that Ampere's law is not violated!

To define an integrated zero tangential H field boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell 3D>Boundaries>Assign>Integrated Zero Tangential H Field**.
   The **Integrated Zero Tangential H Field** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

The new boundary is added to the boundary list in the project tree.

Assigning an Insulating Boundary for the Eddy Current Solver

This boundary condition is generally used to model very thin sheets of perfectly insulating material between touching conductors, as well as infinitely thin cracks inside conductors. Current cannot cross an insulating boundary.

To set an insulating boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Insulating**.
   The **Insulating Boundary** window appears.

3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

Note	An insulating boundary condition can be used to control the flow of induced eddy currents inside conductors that contain sheet objects with an applied insulating boundary condition. In case you apply the insulating boundary condition on sheet objects (inside conductors), always make sure that the respective entities are selected as faces and not as objects before attempting to assign the insulating boundary condition to them.

Assigning a Symmetry Boundary for the Eddy Current Solver

This boundary condition defines a plane of geometric or magnetic symmetry in a structure. Assign it only to the outer surfaces of the problem region.

To set a symmetry boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Symmetry**.
   The **Symmetry Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select one of the following as the type of symmetry:

Odd (Flux Tangential)	H is tangential to the boundary; its normal components are zero.
Even (Flux Normal)	H is normal to the boundary; its tangential components are zero.

5. Optionally, click **Use Defaults** to revert to the default values in the window.
6. Click **OK** to assign the boundary to the selected object.

Warning	When using even symmetry boundaries, always double-check that Ampere's law is not violated!

Related Topics

Setting a Symmetry Multiplier

Assigning an Independent Boundary for the Eddy Current Solver

Independent and dependent boundaries enable you to model planes of periodicity where the H-field at every point on the dependent boundary surface is forced to match the H-field of every corresponding point on the independent boundary surface. The transformation used to map the H-

field from the independent to the dependent is determined by specifying a coordinate system on both the independent and dependent boundaries.

To set an independent boundary:

1. Select the face to which you want to assign the independent boundary.
2. Click **Maxwell>Boundaries>Assign>Matching>Independent**.
   The **Independent Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. In the **Coordinate System** section, do the following to define the coordinate system:

Note	You must specify the coordinate system in the plane on which the boundary exists. First draw the U vector of the coordinate system. Maxwell uses the U vector you draw and the normal vector of the boundary face to calculate the v-axis. Then specify the direction of the V vector.

   a. Select **New Vector** from the **U Vector** pull-down list.
      The **Create Line** message appears, asking you to draw the U vector of the coordinate system in the plane of the selected face, and the **Independent Boundary** dialog box disappears while you draw the U vector.
   b. Select the U vector's origin, which must be on the boundary's surface, in one of the following ways:
      - Click the point.
      - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.
   c. Select a point on the u-axis.
      The **Independent Boundary** dialog box reappears
5. To reverse the direction of the vector, select the **Reverse Direction** check box.
6. Click **OK**.

Maxwell computes the H-field on this boundary and maps it to the dependent boundary using the transformation defined by the independent and dependent coordinate systems.

Assigning a Dependent Boundary for the Eddy Current Solver

Assigning a dependent boundary is the second step in creating matching boundaries. The field on the independent boundary is mapped to the dependent boundary.

Note	You must define a independent boundary before creating the dependent boundaries that are associated with it. Assign dependent boundaries only to the outer surfaces of the problem region.

To set a dependent boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Matching>Dependent**.
   The **Dependent Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.

4. Select the independent boundary to which the dependent is to be assigned from the **Independent** pull-down list.
5. To specify the coordinate system, do the following in the Coordinate System section:

Note	You must specify the coordinate system in the plane on which the boundary exists. First draw the U vector of the coordinate system. Maxwell uses the U vector you draw and the normal vector of the boundary face to calculate the v-axis. Then specify the direction of the V vector.

   a. Select **New Vector** from the **U Vector** pull-down list.
   The **Create Line** message appears asking you to draw the U vector of the coordinate system in the plane of the selected face, and the **Dependent Boundary** dialog box disappears while you draw the U vector.
   b. Select the U vector's origin in one of the following ways:
      - Click the point.
      - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.

Note	The U vector's origin must be on the boundary's surface.

   c. Select a point on the u-axis in the same way.
   When the second point is identified, the vector is set, and the **Dependent Boundary** window reappears, with **Defined** listed in the **U Vector** pull-down list.
   d. To specify the direction of the V vector, select or clear the **Reverse Direction** check box.

Note	When entering current, current density terminals, or coil terminals, the arrow associated with current in the selected object shows the direction as if a positive value is entered for the current. If a negative value is entered the actual current flow direction is opposite to what the arrow shows. Click **Reverse Direction** if you want to reverse the direction of the arrow.

6. In the **Relation** section, select one of the following to specify the field behavior on the boundary:

Hdep = Hind	Click this radio button if the dependent and independent boundaries have the same magnitude and direction.
Hdep = -Hind	Click this radio button if the dependent boundary field has the same magnitude as but the opposite direction from the independent boundary field.

7. Optionally, click **Use Defaults** to revert to the default values in the window.
8. Click **OK** to assign the boundary to the selected object.

Note	The origin must be a vertex point of one of the objects.

Assigning a Radiation Boundary for the Eddy Current Solver

To simulate problems that allow fields to radiate infinitely far into space, you can define surfaces to be radiation boundaries. The system absorbs the field at the radiation boundary, essentially ballooning the boundary infinitely far away from the structure. In a far field region, field components are expressed by:

$$\boldsymbol{E} = -Zn \times \boldsymbol{H}$$

where:

$$Z = \sqrt{\frac{\mu}{\varepsilon}}$$

and Z is the component of the E-field that is tangential to the surface.

Using the field impedance, the equation becomes:

$$n \times \boldsymbol{E} = -Zn \times (n \times \boldsymbol{H})$$

which is used as a radiation boundary. The radiation boundary condition should be placed far enough from the source of radiation so that the approximation of the far field in the region of the boundary holds.

The second-order radiation boundary condition is an approximation of free space. The accuracy of the approximation depends on the distance between the boundary and the object from which the radiation emanates.

A radiation surface does not have to be spherical. However, it should be exposed to the background, convex with regard to the radiation source, and located at least one-quarter of a wavelength away from the radiating sources. In some cases you may want to use smaller distances.

To assign a radiation boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).

2. Click **Maxwell>Boundaries>Assign>Radiation**.
   The **Radiation Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Optionally, click **Use Defaults** to revert to the default values in the window.
5. Click **OK** to assign the boundary to the selected object.

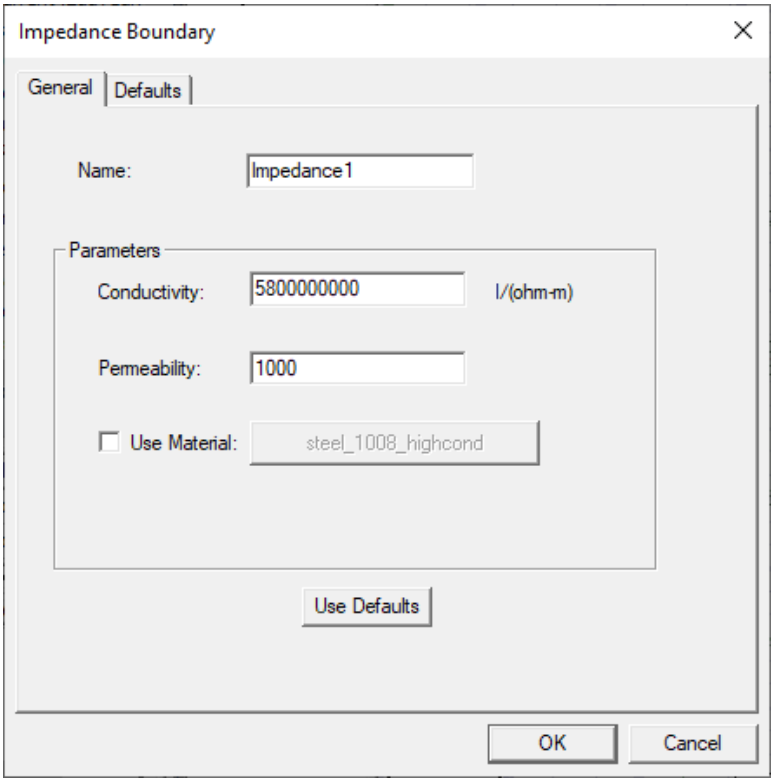
Assigning an Impedance Boundary for the Eddy Current Solver

> **Note:** To define an impedance boundary for the transient solver, refer to *Assigning an Impedance Boundary for the Transient Solver*.

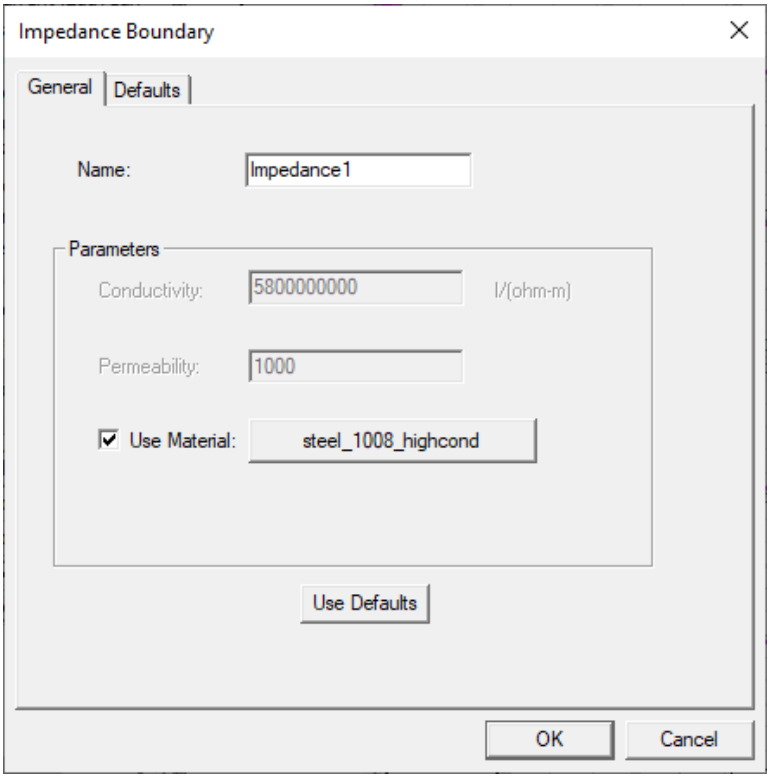
This boundary condition is used to simulate the effect of induced currents in a conductor without explicitly computing them. Since the conductor must be excluded from the model (saving time needed to mesh and solve for currents), assign the impedance boundary condition to an outside edge of the problem region or to an excluded object.

To define an impedance boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell 3D>Boundaries>Assign>Impedance** to open the **Impedance Boundary** dialog box.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Do one of the following:
   - If the **Use Material** check box is unchecked, enter the conductivity (in inverse ohm-meters) in the **Conductivity** field, and the conductor's relative permeability in the **Permeability** field.



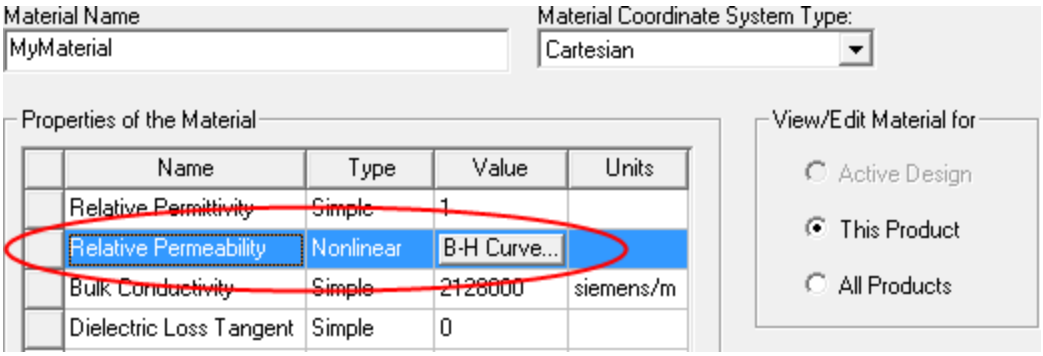
- If you wish to select a material from the library, check the **Use Material** check box, then use the button to select the desired material.



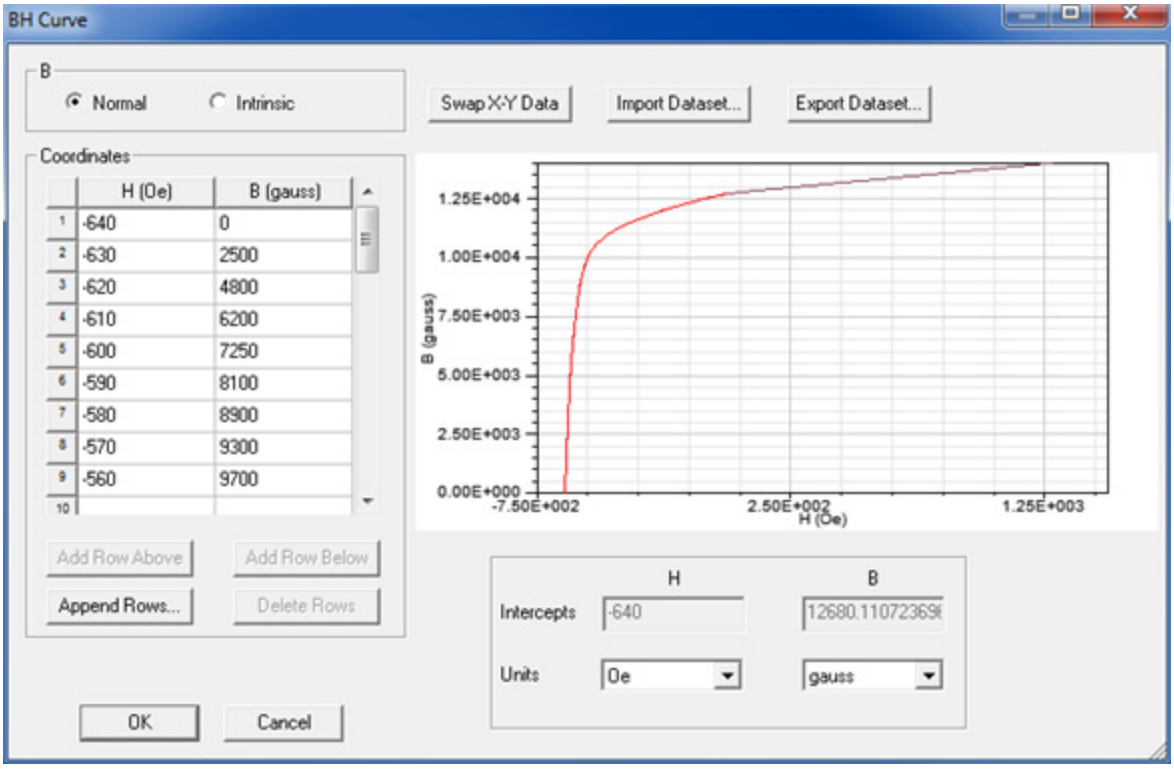
- **Nonlinear Permeability**

  If you select a material that has nonlinear permeability, the non-linear coefficient is computed automatically.

  The nonlinear relative permeability of the material is determined by the B-H curve. If the desired B-H curve of the permeability belongs to an existing material, you can directly use that existing material.



  Otherwise, you can edit a new material with the desired B-H curve.



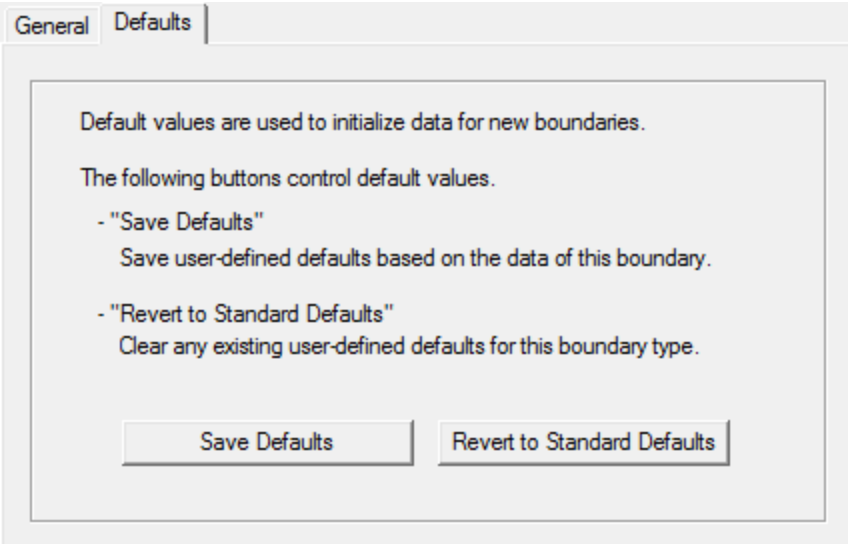
Refer to *Relative Permeability for a Maxwell or RMxprt Material* for details on using nonlinear relative permeability.

- **Temperature Dependent Impedance Boundary**

  The electrical conductivity and permeability of a material used for an impedance boundary condition can be temperature dependent. The Maxwell 3D eddy solver provides an option to allow you to input functions as thermal modifiers to consider the effect of temperature on those properties. To enable this feature, check the **Use Material** check box as above. You can then click **Material** to open the material properties panel in which you can add a thermal modifier. (Refer to *Setting the Temperature of Objects* for additional information on using temperature dependent impedance boundaries).

5. Optionally, click **Use Defaults** to revert to the default values in the window.

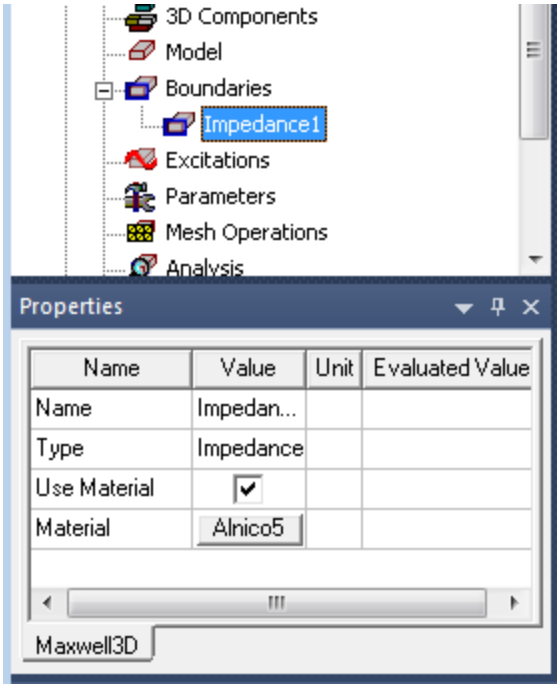
The **Defaults** tab allows you to control default values. The **Save Defaults** button saves the values currently defined on the **General** tab as the defaults to be assigned to new impedance boundaries. **Revert to Standard Defaults** clears existing user-defined values and replaces them with the standard default values.



6. Click **OK** to assign the boundary to the selected object.

   The Project Manager lists the newly assigned impedance boundary in the tree. You can select the boundary in the tree to view and edit its properties in the Properties Window. You

can also double-click the boundary entry in the tree to open it for editing in the Impedance Boundary dialog box.



Related Topics

Impedance Boundary

Setting a Thermal Modifier

Setting the Temperature of Objects

Impedance Boundary

Impedance boundaries allow you to simulate the effect of induced currents in a conductor without explicitly computing them. Use this boundary condition for models where:

- The skin depth in the conductor is less than two orders of magnitude smaller than the dimensions of the structure. In models like this, the meshmaker may not be able to create a fine enough mesh in the conductor to compute eddy currents.
- The magnetic field decays much more rapidly inside the conductor in the direction that's normal to the surface than it does in directions that are tangential to the surface.
- The AC current source is relatively far away from the surface where eddy currents occur, compared to the size of the skin depth.

The conductor itself must be excluded from the solution region. When setting up the model, do one of the following:

- For external boundaries, when drawing the model, make the surface along which eddy currents are to be computed an outer surface of the problem region.

- For internal impedance boundaries, assign perfect conductor to the object in the Material Manager or uncheck **Solve Inside** to exclude the object from the solution domain.

Then, when defining boundaries, assign an impedance boundary to the individual surfaces of the problem region (for external boundaries) or to the entire object (for internal boundaries). By entering the conductivity, s, and the relative permeability, mr, of the object, you specify the skin depth of induced eddy currents. The simulator uses this skin depth value when computing the electromagnetic field solution. It assumes that the H-field falls off exponentially inside the conductor. The ohmic loss due to induced currents can then be computed from the tangential components of the H-field along the impedance boundary — the surface of the object that you are interested in.

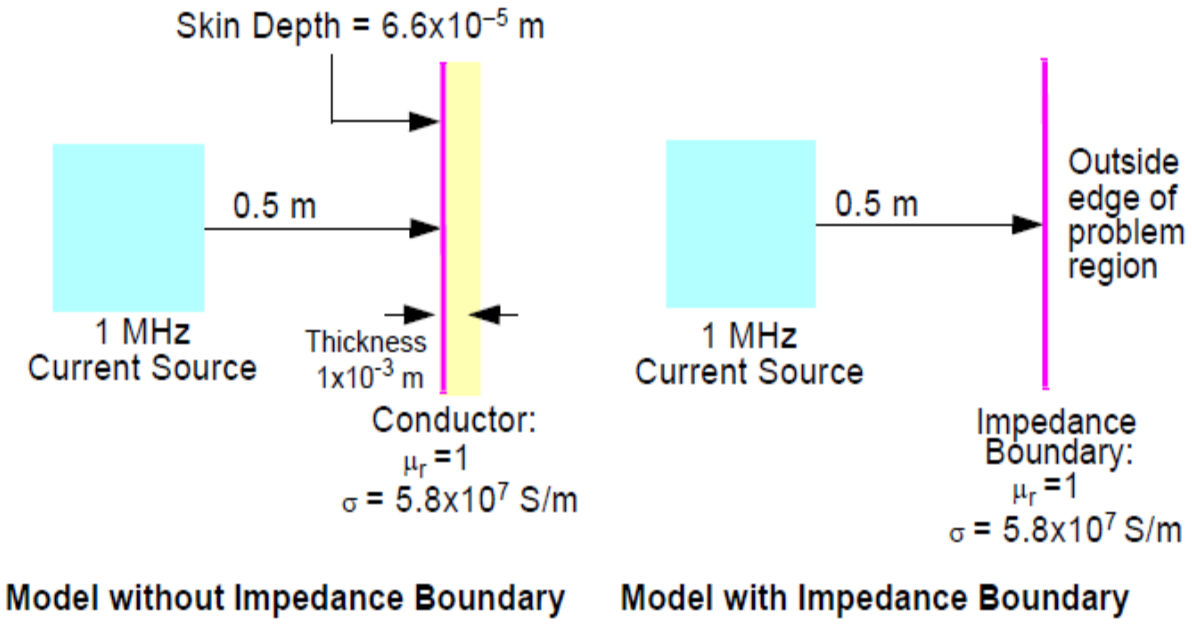
Note	An impedance boundary only approximates the effect of eddy currents acting at a shallow skin depth. It does not directly compute them. In general, the fields modeled using an impedance boundary will closely match the field patterns that would actually occur in the structure. However, at discontinuities in the surface (such as corners), the field patterns may be different.
Note	For the current version of the Maxwell solver, when a project includes all three of the following at the same time: an impedance boundary condition, a material with anisotropic permeability, and a winding, the project cannot be solved by the Maxwell eddy current solver and the Maxwell 3D transient solver.

Related Topics

When to use Impedance Boundaries

When to Use Impedance Boundaries

A typical situation where impedance boundaries can be used to reduce the complexity of a model is shown below. Suppose you want to compute eddy current losses in the conductor next to the current source shown below on the left. If the source carries AC current at a frequency of 1 MHz, the skin depth in the conductor is 6.6 x 10–5 meters. This is several orders of magnitude smaller than the conductor's thickness. Since the conductor where currents are induced is also relatively far away from the current source, an impedance boundary can be used to model the induced currents — as shown on the right.



The conductor itself must be excluded from the model. Instead, the outside boundary of the model is moved to the inside surface of the conductor. This outside surface is defined as an impedance boundary, using the conductivity and permeability specified previously. Since the simulator does not have to actually compute a solution inside the conductor, the field solution is computed more quickly and uses less memory. After solving, you can compute the ohmic loss for the surface using the solution calculator and plot the loss density on the boundary.

Assigning a Resistive Sheet Boundary for the 3D Eddy Current Solver

Note	For Transient designs refer to *Assigning a Resistive Sheet Boundary for the Transient Solver*.

When two conductors are in contact within a conduction path, a voltage drop normally exists across the contact surface due to the imperfection of the contact between the two conductors. This boundary condition assigns a resistive sheet to consider the impact of this kind of voltage drop. The resistive sheet is defined by a lumped resistance in ohms.

The following are usages of the resistive sheet boundary:

- **Typical usage**: Modeling contact resistance by applying Resistive Sheet boundary to a contacting surface (face) of two conductive objects.
- **Other possible usages**: Resistive Sheet boundary can be applied to any sheet object inside a conductive object or on a conductive object surface in order to increase the total resistance of the conducting path and to introduce localized loss density to the place of the sheet location.

This boundary can also be used to model terminal contact resistance (for stranded or solid conductors). For example, by applying it to the conductor cross section touching the Region. In this case the Excitation Terminal and Resistive Sheet boundary are not allowed to be on the same sheet object.

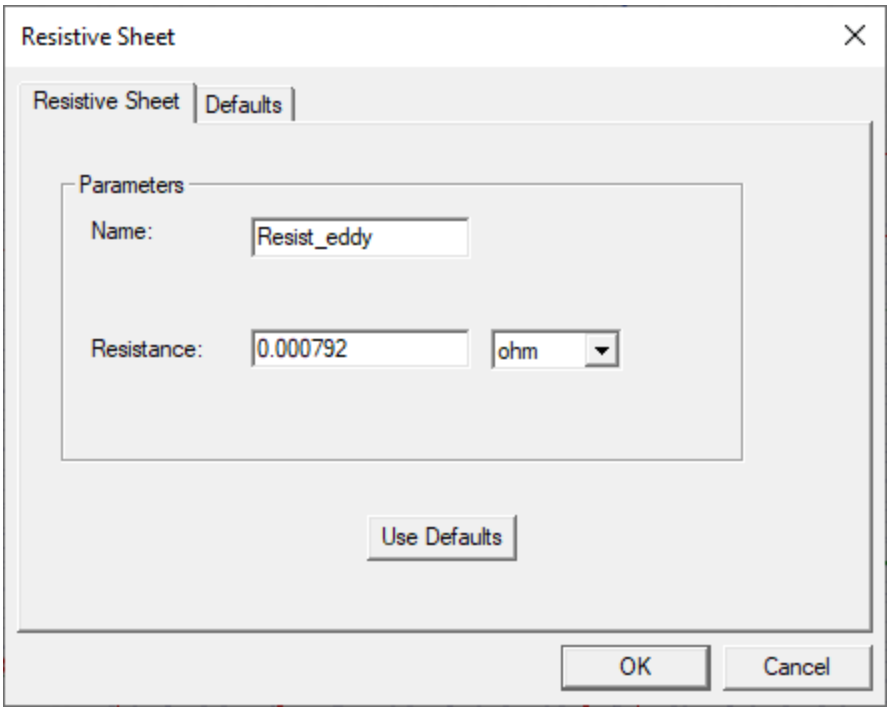
The losses incurred due to a Resistive Sheet boundary are output as Surface Loss Density. These losses are counted as losses in solid conductors and are reported together with the losses in other solid conductors under Solid Loss. If the Surface Loss Density is mapped to Mechanical, it should be assigned to a face as Heat Flux.

To define a resistive sheet boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a sheet object or face of a 3D conducting body touching another conducting body or face of a 3D conducting body touching the Region).

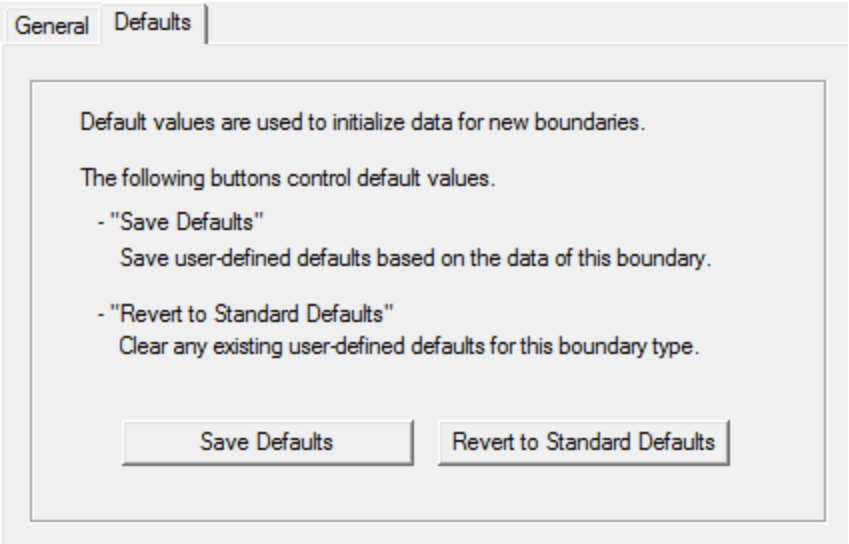
Note	• The sheet must be completely inside a conductor or on its surface. • The sheet must have conductors touching both of its faces. • The resistance value cannot be dependent on intrinsic variable.

2. Click **Maxwell 3D>Boundaries>Assign>Resistive Sheet** to open the **Resistive Sheet** dialog box.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Enter the resistance value in the **Resistance** field, and select a unit of measure. Use of variables is supported.



5. Optionally, click **Use Defaults** to revert to the default value (1 ohm) in the window.

   The **Defaults** tab allows you to control default values. The **Save Defaults** button saves the values currently defined on the **General** tab as the defaults to be assigned to new impedance boundaries. **Revert to Standard Defaults** clears existing user-defined values and replaces them with the standard default values.



6. Click **OK** to assign the boundary to the selected object.

   The Project Manager lists the newly assigned resistive sheet boundary in the tree. You can select the boundary in the tree to view and edit its properties in the Properties Window. You can also double-click the boundary entry in the tree to open it for editing in the **Resistive Sheet** dialog box.

> **Note:** When switching the solution type between transient and eddy current, resistive sheet boundaries will be kept.

Related Topics

Assigning a Resistive Sheet Boundary for the Transient Solver

Eddy Current Excitations

The eddy current solver allows you to define the following sources of AC magnetic fields:

Excitation	Type of Excitation
Current Density	The current density in a conductor. In this case, you must also define the current density terminals for the conductor.

Current Density Terminal	Specifies the cross-section of the conductor where current density is specified, to be used as a terminal object. Current Density and Current Density Terminal are required to be used in conjunction.
Current	The total current in a conductor.
Coil Terminal	Used to define one or more model windings.
Winding With Current	Current for both a stranded and solid conductor.
Winding With Voltage	Voltage for both a stranded and solid conductor.
Winding With External Circuit Connection	External circuit connection for both a stranded and solid conductor.

Note	When entering current, the arrow associated with current in the selected object shows the direction as if a positive value is entered for the current. If a negative value is entered the actual current flow direction is opposite to what the arrow shows. Click **Reverse Direction** if you want to reverse the direction of the arrow.

Note	When a source conductor touches an outer boundary, Maxwell assumes current will flow perpendicular to the boundary. If you do not want current to cross the outer boundary, an insulating boundary should be used.

Assigning a Current Density Excitation for the Eddy Current Solver

This command specifies the x-, y-, and z-components of the AC current density in a conduction path. If the current density is a function of position, the value is entered in ampere/m^2, even if you change the units in the problem.

To define the current density:

1. Select the section of the geometry (i.e., the conductor) in which you want to apply the excitation (typically a 3D object).
2. Click **Maxwell>Excitations>Assign>Current Density**.
   The **Current Density Excitation** window appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, specify the following options:
   a. Enter the **X Component**, **Y Component**, and **Z Component** values to define the respective components of the current density vector.
   b. Do the following in the two **Coordinate System** pull-down lists:

- From the first pull-down list, select either **Global** or **RelativeCS1** (where x = 1,2,3… if any relative coordinate systems have been defined).
- From the second pull-down list, select **Cartesian**, **Cylindrical**, or **Spherical**.

c. Enter the phase angle, θ, of the current density in the **Phase** field.

5. Optionally, click **Use Defaults** to revert to the default values in the window.
6. Click **OK** to assign the excitation to the selected object.

To complete the current density definition, you also need to specify any related current density terminals, which must be assigned to a 2D object or face of the respective conduction path.

Note	Select the names of the sheet objects to serve as the current density terminals. This object must form an exact cross-section of the current density conduction path.You must create 2D objects which represent locations where current flows into and out of the problem region, or branches at any location in the conduction path. In current loops, any exact 2D cross-section may serve as a terminal.

Related Topics

Assigning a Current Density Terminal

Assigning a Current Density Terminal Excitation for the Eddy Current Solver

To define the current density terminal:

1. Select the section of the geometry (i.e., the conductor) in which you want to apply the excitation (typically a 3D object).
2. Click **Maxwell>Excitations>Assign>Current Density Terminal**.
   The **Current Density Terminal Excitation** window appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. Click **OK** to assign the excitation to the selected target(s).

Related Topics

Assigning a Current Density Excitation

Assigning a Current Excitation for the Eddy Current Solver

Specifies the total AC current in a conduction path. The conduction path may be contained completely within the problem region (for example, a coil), or may touch the edges of the problem region.

To specify the total AC current:

1. Select the 2D section of the geometry on which you want to apply the excitation.
2. Click **Maxwell>Excitations>Assign>Current**.
   The **Current Excitation** window appears.

3. Enter a name for the excitation in the **Name** box, or accept the default.
4. Enter the current strength in the **Value** box.
5. Enter the phase for the current in the **Phase** box.
6. Select either **Solid** or **Stranded** to define the **Type** for the conductor. For a stranded conductor, a uniformly distributed current density is assumed without considering eddy effects.
7. Optionally, click **Swap Direction** to change the direction of the current flow.
8. Optionally, click **Use Defaults** to revert to the default values in the window.
9. Click **OK** to assign the excitation to the selected object.

Assigning a Coil Terminal for an Eddy Current Solver

To assign a coil terminal as an Eddy current excitation:

1. Select the section of the geometry on which you want to apply the excitation (typically a 2D planar object).
2. Click **Maxwell>Excitations>Assign>Coil Terminal**. (You can also right-click an existing winding in the project tree, and select **Assign Coil Terminal**.)

   The **Coil Terminal Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the number of conductors for the coil terminal in the **Number of Conductors** box. For a solid winding, the number should be 1.

Note	This value represents only the number of conductors inside of the selected geometry. If the Coil Terminal is cut due to symmetry, then only enter the Number of Conductors in the portion modeled.

5. If you want to switch the direction for the coil terminal, click **Swap Direction**.

Note	The direction is shown in the model by a bold red arrow.

6. Optionally, click **Use Defaults** to revert to the default values in the window.
7. Click **OK** to assign the coil terminal to the selected object.

Note	To add a coil terminal to an existing winding, do the following: 1. In the project tree, right-click the coil terminal, and select **Add to Winding**. The **Add to Winding** dialog box appears. 2. Select the winding to which you ant to add the coil terminal, and click **OK**.

Note	To remove a coil terminal from an existing winding, do the following: • In the project tree, right-click the coil terminal, and select **Remove from Winding**. The terminal is removed from the winding and moved up one level

	in the project tree (directly beneath **Excitations**).

Assigning a Winding Setup for an Eddy Current Solver

Note	• If you wish to define a winding for a Transient solution type, refer to "Assigning a Winding Setup for a Transient Solver " on page 11-101 . • If you wish to define a winding for a Magnetostatic solution type, refer to Assigning a Winding Setup for a Magnetostatic Solver.

You can use the pre-defined coil terminal(s) to define one or more current or voltage windings.

To define a winding for your model:

1. Click **Maxwell>Excitations>Add Winding**. to open the **Winding** dialog box.
2. Enter a name for the winding in the **Name** box, or accept the default.
3. In the **Parameters** section:
   a. Select **Current**, **Voltage**, or **External** from the **Type** pull-down list. The winding type selected enables only those fields (Initial Current, Resistance, Inductance, Voltage, Phase) applicable for the chosen type.
   b. Select the **Solid** or **Stranded** radio button to specify the type of conductor.
   c. Enter values in the following fields (enabled according to winding type), and select the desired units. You can also type a function as an expression for any of these fields except Initial Current.

For a current winding	For a voltage winding[1]	For an external winding
Current	Resistance	(no settings required)
Phase	Inductance	
	Voltage	
	Phase	

   d. Enter a value in the **Number of parallel branches** text box.

Note	Parallel branches for solid windings assume there are no circulating currents in the parallel branches.

4. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
5. Optionally, on the Defaults tab, click **Save Defaults** to save the current settings or click **Revert to Standard Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the excitation to the selected object.

A winding is now defined for your model.

Note	For an external winding, you must also set up an external circuit connection.
Note	To add a coil terminal to this winding, do the following:

	1. In the project tree, right-click the winding, and select **Add Terminals**. The **Add Terminals** dialog box appears, listing all coil terminals that do not already belong to that winding. 2. Select the coil terminal (s) you want to add. To select multiple terminals, press **CTRL** and click each terminal. 3. Click **OK**. To assign a coil terminal excitation and add it to this winding, do the following: 1. Select the section of the geometry on which you want to apply the coil terminal excitation. 2. In the project tree, right-click the winding, and select **Assign Coil Terminal**. The **Coil Terminal Excitation** dialog box appears. 3. Click **OK** to assign the coil terminal excitation. The coil terminal excitation is assigned and is added to the winding. To delete all coil terminal excitations that belong to this winding, do the following: • In the project tree, right-click the winding, and select **Delete All Terminals**. All coil terminal excitations are removed from the winding and deleted from the excitations.

---

1. For a solid winding, the resistance term can represent: the resistance of a portion of the winding which is not modeled (for instance, end-effects), the leads connecting the winding to the source, or the source resistance. (The main winding resistance is calculated directly by the solver.) For a stranded winding, the resistance term is the complete DC resistance of the winding (since the solver does not determine resistance of a stranded winding) as well as the resistance of the end-effects, leads, source, etc. For both solid and stranded windings, the inductance term can represent: the extra inductance for a portion of the winding of the winding which is not modeled (for instance, end-effects), the leads connecting the winding to the source, or the source inductance. (The main winding inductance itself is calculated directly by the solver.)

External Circuit Setup for 3D Eddy Current

The Maxwell Circuit design editor is used to set up the external circuit excitation for windings in Maxwell 2D and 3D Eddy current solutions. Because the Eddy current solver is a frequency domain solver, only linear circuit components are supported, which include Resistance, Inductance, Capacitance, and Transformer. In addition, only sinusoidal sources should be used as sources. Sinusoidal Current Source and Sinusoidal Voltage Source, designed for transient solutions, can also be used for eddy current solutions. For example, to use the sinusoidal current source, you should set the parameters as follows:

- Set **I0**, **Td**, and **Df** to zero, where I0 is offset current in Amps, Td is the delay time in seconds, and Df is the damping factor in 1/seconds.
- Set **TIME** as the **Type**.

- Use **Ia** to set the peak amplitude in Amps.
- Use **IFreq** to set the signal frequency. Note that setting IFreq is not necessary as it will be internally set to the frequency defined in the Maxwell design setup.
- Use **Phase** to set the phase delay.

With these specifications, the time signal can be written as:

$$I(t) = I_a \sin\left(2\pi I_{freq} - Phase\right)$$

Note that “Phase” is the signal phase delay. The complex source in the eddy current solver is represented in phasor form as:

$$I = I_a \ e^{j\theta}$$

and the time signal can be written in terms of (I_a, θ) as:

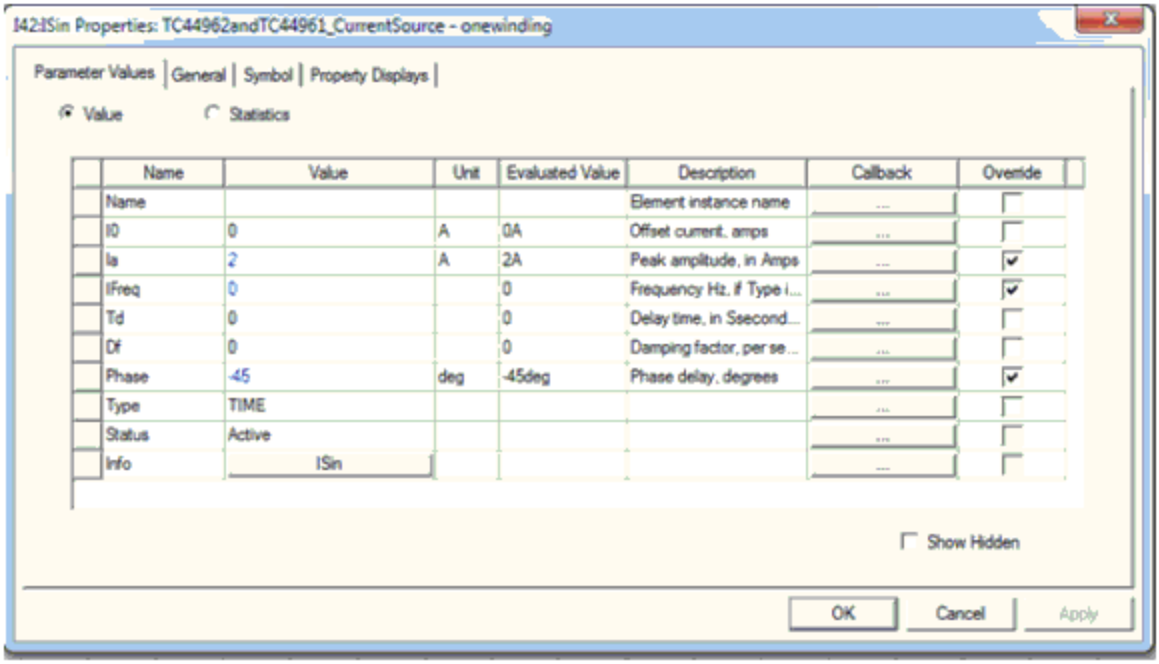
$$I(t) = Im\left\{I_a \ e^{j(2\pi I_{freq}+\theta)}\right\}$$

In the complex expression, θ is defined as phase, while in Maxwell Circuit, *Phase* is defined as phase delay. Thus:

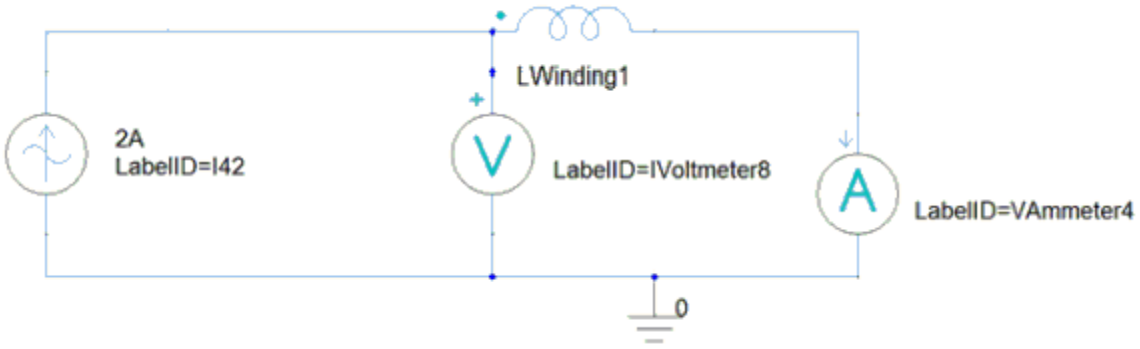
$$Phase = -\theta$$

As a result, you should enter the negative of phase of the complex signal as the Phase value.

An example of such a setup for $I = 2e^{j45^\circ}$ is shown below.



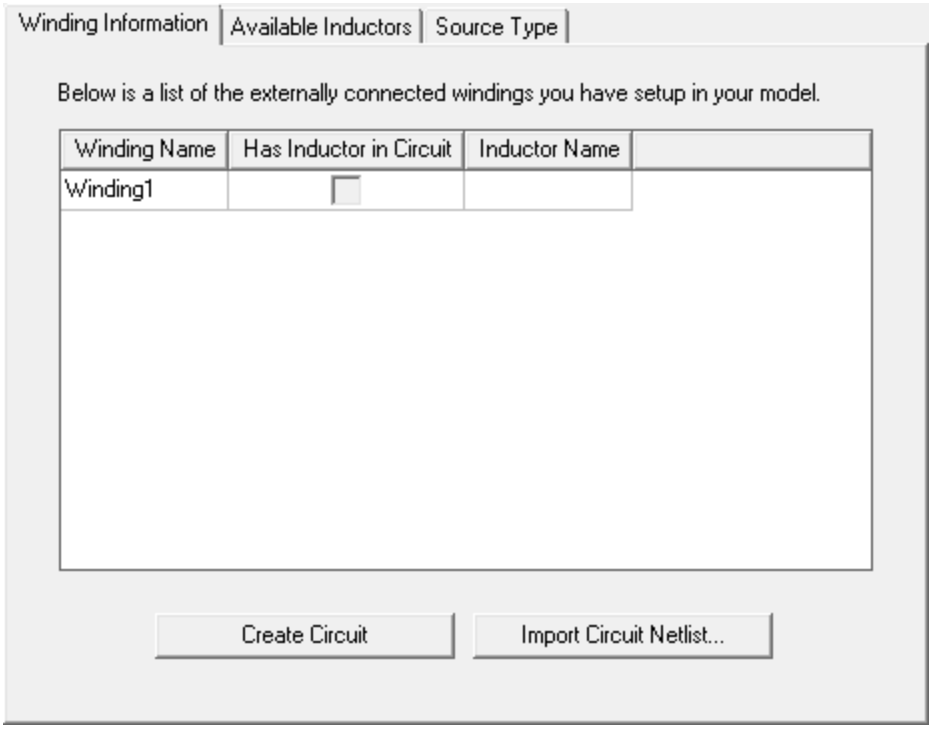
An example of an external circuit excitation for the winding for an Eddy current solver is shown below.



Note	For frequency domain eddy current solutions, only linear circuit components are supported, which include Resistor, Inductor, Capacitor, and Transformer. Only sinusoidal sources should be used as sources.

To set up an external circuit connection for an External winding type:

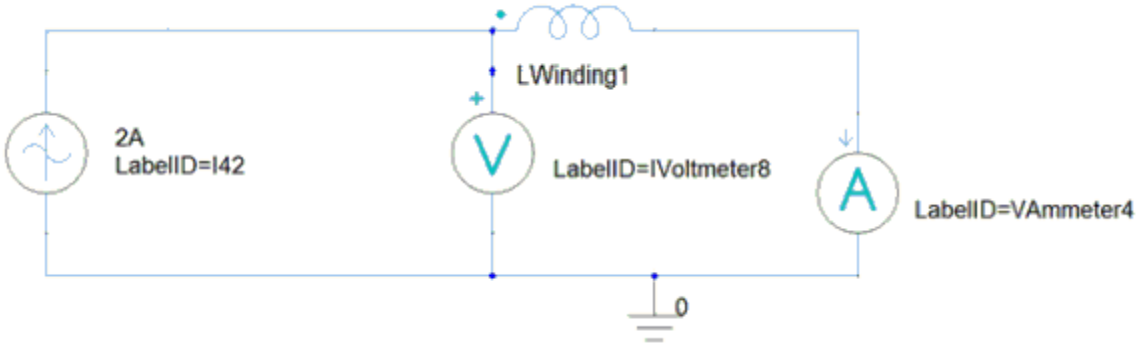
1. Click **Maxwell 3D>Excitations>External Circuit>Edit External Circuit** to open the **Edit External Circuit** dialog box. You can also right-click on **Excitations** in the Project tree and select **External Circuit>Edit External Circuit**.
    a. If no external winding connections have been set up for your model, the **Winding Information** tab lists the external winding(s) in the design. Continue with step 2.



b. If external windings have already been set up, continue with step 7.

2. On the **Winding Information** tab, click **Create Circuit** to add a circuit design to the project. The new circuit design automatically opens for editing in the Maxwell Circuit Editor. The circuit includes one inductor for each external winding in the original Maxwell transient design. The following example is for a design having one external winding.

3. Add the components needed to complete the external circuit design such as shown the following example.



4. When the circuit design is finished, click **Maxwell Circuit>Export Netlist** and export the netlist **.sph** file to the desired location.
5. Return to the Maxwell design and click **Maxwell 3D>Excitations>External Circuit>Edit External Circuit** to open the **Edit External Circuit** dialog box. You can also right-click on **Excitations** in the Project tree and select **External Circuit>Edit External Circuit**.
6. On the **Winding Information** tab, click **Import Circuit** to import the circuit.
7. To view a list of inductors in the imported circuit, click the **Available Inductors** tab.
8. To view the sources used in the externally connected windings, click the **Source Type** tab.
9. If variables are used in the imported circuit, they and their design values are listed on the **Parameter Values** tab. You can link to variables within the imported circuit as follows:
    a. On the **Parameter Values** tab, click in the **Value** column of the parameter to be mapped.

b. Enter a new local variable name and press the `Enter` key to open the **Add Variable** dialog.

c. Select a **Unit Type** and **Unit**, and enter a **Value** for the variable and click **OK**. The new Maxwell variable is now mapped to the variable in the imported circuit, and can be varied directly by Maxwell or used for Optimetrics analyses.

10. If the imported file is a .**sph** file, you can click the **Circuit Path** tab to view the original project and design names.

Note	After you import a circuit netlist, the **Create Circuit** button changes to **Update Netlist from Circuit**. Use **Update Netlist from Circuit** to reimport the netlist to keep it current.

11. Click **OK** to close the **Edit External Circuit** dialog box.

Related Topics

"Assigning a Winding Setup for an Eddy Current Solver " on page 11-81

Transient Boundaries and Excitations

Define at least one of the following as a source of magnetic fields:

- The stranded or solid windings, with a voltage or current supply or connected to an external circuit winding.

Note	For stranded conductors, it is assumed that the current density is uniform on the cross-section of the respective conductors. Thus, the stranded option for current excitations should mostly be used only in cases where the cross-section of conductors is constant, consistently with the assumption that the respective object is a coil built with strands of wire. For this reason, conductors (coils) could be created by either sweeping around an axis using zero segments or sweeping along a path having only true surface segments in order to create a smooth surface coil having a completely uniform cross-sectional area over its entire length.

- A permanent magnet.

You may need to set at least one outer boundary to the following:

- The default boundary conditions.
- An odd symmetry (flux tangential) boundary.
- An even symmetry (flux normal) boundary.

Related Topics

Technical Notes: 3D Transient Excitations (Sources)

Specifying the Solver Type

Transient Boundaries

The transient field solver allows you to define the following types of boundary conditions.

Boundary Type	H-Field Behavior	Used to model...
Default Boundary Conditions (Natural and Neumann)	Field behaves as follows: • Natural boundaries — **H** is continuous across the boundary. • Neumann boundaries — **H** is tangential to the boundary and flux cannot cross it.	Ordinary field behavior. Initially, object interfaces are natural boundaries; outer boundaries, and excluded objects are Neumann boundaries.
Zero Tangential H Field	The tangential components of **H** are set to zero. Flux is perpendicular.	External magnetic fields.
Integrated Zero Tangential H Field	For applications such as a motor with the shaft excluded, there remains a hole in the middle. In such cases, on the hole's boundary, neither the Tangential H Field boundary (where integration of the tangential H field is non-zero), nor the Zero Tangential H Field boundary (where the tangential H field is zero everywhere) can be applied because – for the hole boundary case – integration of the tangential H field is zero, but the tangential H field is not zero everywhere. Integrated Zero Tangential H Field can be used for such applications.	Cases such as the boundary around the hole that remains when a motor's shaft is excluded.
Insulating	Same as Natural boundary, except that current cannot cross the boundary.	Thin, perfectly insulating sheets between touching conductors.
Symmetry	Field behaves as follows: • Odd Symmetry (Flux Tangential) — **H** is tangential to the boundary; its normal components are zero. • Even Symmetry (Flux Normal) — **H** is normal to the boundary; its tangential components are zero.	Planes of geometric and magnetic symmetry.

Matching (Independent and Dependent)	The H-field on the dependent boundary is forced to match the magnitude and direction (or the negative of the direction) of the H-field on the independent boundary.	Planes of symmetry in periodic structures where **H** is oblique to the boundary.
Resistive Sheet	Same as Natural boundary.	Voltage drop and loss on the contact surface between two different objects within a conduction path.
Impedance	Includes the effect of induced currents beyond the boundary surface.	Conductors with very small skin depths.
Touching	Same as Natural boundary.	Interface between a moving part and a stationary part.

Default Boundary Conditions for a Transient Solver

These boundary conditions are automatically defined for a transient model:

- Natural boundaries are assigned to the surfaces between objects.
- Neumann boundaries are assigned to the outside edges of the problem region.

To leave a surface set to its default boundary condition, you do not need to set any boundary conditions. Deleted boundary conditions and excitations automatically reset to the default boundary conditions.

Assigning a Zero Tangential H Field Boundary for a Transient Solver

This type of boundary defines external magnetic fields in a model.

Warning	When using zero tangential magnetic field boundary conditions, always double-check that Ampere's law is not violated!

To define a zero tangential H field boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).

2. Click **Maxwell>Boundaries>Assign>Zero Tangential H Field**.
   The **Zero Tangential H Field** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

Assigning an Integrated Zero Tangential H Field Boundary for a Transient Solver

This type of boundary is applied when the integral of the tangential H field is zero.

Warning	When using integrated zero tangential magnetic field boundary conditions, always double-check that Ampere's law is not violated!

To define an integrated zero tangential H field boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell 3D>Boundaries>Assign>Integrated Zero Tangential H Field**.
   The **Integrated Zero Tangential H Field** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

Assigning an Insulating Boundary for a Transient Solver

This boundary condition is used to model very thin sheets of perfectly insulated material between touching conductors. Current cannot cross an insulating boundary.

To set an insulating boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Insulating**.
   The **Insulating Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

Note	An insulating boundary condition can operate on the source current. It can also be used to control the flow of induced eddy currents.

Assigning a Symmetry Boundary for a Transient Solver

This boundary condition defines a plane of geometric or magnetic symmetry in a structure. Assign it only to the outer surfaces of the problem region.

To set a symmetry boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).

2. Click **Maxwell>Boundaries>Assign>Symmetry**.
   The **Symmetry Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select one of the following as the type of symmetry:

Odd (Flux Tangential)	H is tangential to the boundary; its normal components are zero.
Even (Flux Normal)	H is normal to the boundary; its tangential components are zero.

5. Optionally, click **Use Defaults** to revert to the default values in the window.
6. Click **OK** to assign the boundary to the selected object.

Warning	When using even symmetry boundaries, always double-check that Ampere's law is not violated!

Related Topics

Setting a Symmetry Multiplier

Assigning a Independent Boundary for a Transient Solver

Independent and dependent boundaries enable you to model planes of periodicity where the H-field at every point on the dependent boundary surface is forced to match the H-field of every corresponding point on the independent boundary surface. The transformation used to map the H-field from the independent to the dependent is determined by specifying a coordinate system on both the independent and dependent boundaries.

To set an independent boundary:

1. Select the face to which you want to assign the independent boundary.
2. Click **Maxwell>Boundaries>Assign>Matching>Independent**.
   The **Independent Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. In the **Coordinate System** section, do the following to define the coordinate system:

Note	You must specify the coordinate system in the plane on which the boundary exists. First draw the U vector of the coordinate system. Maxwell uses the U vector you draw and the normal vector of the boundary face to calculate the v-axis. Then specify the direction of the V vector.

   a. Select **New Vector** from the **U Vector** pull-down list.
      The **Create Line** message appears, asking you to draw the U vector of the coordinate system in the plane of the selected face, and the **Independent Boundary** dialog box disappears while you draw the U vector.

b. Select the U vector's origin, which must be on the boundary's surface, in one of the following ways:
   - Click the point.
   - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.

c. Select a point on the u-axis.
   The **Independent Boundary** dialog box reappears

5. To reverse the direction of the vector, select the **Reverse Direction** check box.
6. Click **OK**.

Maxwell computes the H-field on this boundary and maps it to the dependent boundary using the transformation defined by the independent and dependent coordinate systems.

Assigning a Dependent Boundary for a Transient Solver

Assigning a dependent boundary is the second step in creating matching boundaries. The field on the independent boundary is mapped to the dependent boundary.

Note	You must define an independent boundary before creating the dependent boundaries that are associated with it. Assign dependent boundaries only to the outer surfaces of the problem region.

To set a dependent boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell>Boundaries>Assign>Matching>Dependent**.
   The **Dependent Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select the independent boundary to which the dependent is to be assigned from the **Independent** pull-down list.
5. To specify the coordinate system, do the following in the Coordinate System section:

Note	You must specify the coordinate system in the plane on which the boundary exists. First draw the U vector of the coordinate system. Maxwell uses the U vector you draw and the normal vector of the boundary face to calculate the v-axis. Then specify the direction of the V vector.

   a. Select **New Vector** from the **U Vector** pull-down list.
      The **Create Line** message appears asking you to draw the U vector of the coordinate system in the plane of the selected face, and the **Dependent Boundary** dialog box disappears while you draw the U vector.
   b. Select the U vector's origin in one of the following ways:
      - Click the point.
      - Type the point's coordinates in the in the **X**, **Y**, and **Z** boxes.

Note	The U vector's origin must be on the boundary's surface.

c. Select a point on the u-axis in the same way.
When the second point is identified, the vector is set, and the **Dependent Boundary** window reappears, with **Defined** listed in the **U Vector** pull-down list.

d. To specify the direction of the V vector, select or clear the **Reverse Direction** check box.

6. In the **Relation** section, select one of the following to specify the field behavior on the boundary:

Hdep = Hind	Click this radio button if the dependent and independent boundaries have the same magnitude and direction.
Hdep = -Hind	Click this radio button if the dependent boundary field has the same magnitude as but the opposite direction from the independent boundary field.

7. Optionally, click **Use Defaults** to revert to the default values in the window.
8. Click **OK** to assign the boundary to the selected object.

Note	The origin must be a vertex point of one of the objects.

Assigning a Resistive Sheet Boundary for the Transient Solver

Note	For Eddy Current designs refer to *Assigning a Resistive Sheet Boundary for the Eddy Current Solver*.

When two conductors are in contact within a conduction path, a voltage drop normally exists across the contact surface due to the imperfection of the contact between the two conductors. This voltage drop should be taken into account when the winding is a voltage excitation. This boundary condition assigns a resistive sheet to consider the impact of this kind of voltage drop. The resistive sheet is defined by a lumped resistance in ohms.

To assign a resistive sheet boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a sheet object or face of a 3D conducting body touching another conducting body).

Note	• The sheet must have conductors touching both of its faces. • The sheet can not be smaller than the contacting surface. • The sheet cannot span multiple faces.

2. Click **Maxwell>Boundaries>Assign>Resistive Sheet**.
The **Resistive Sheet** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Enter a resistance value in the **Resistance** box and select the unit of measure, or accept the default (1 ohm).

5. Optionally, the **Defaults** tab allows you to save the current resistance value for future use, and to revert to the standard defaults.
6. Click **OK** to assign the boundary to the selected object.

> **Note:** When switching the solution type between transient and eddy current, resistive sheet boundaries will be kept.

Related Topics

Assigning a Resistive Sheet Boundary for the Eddy Current Solver

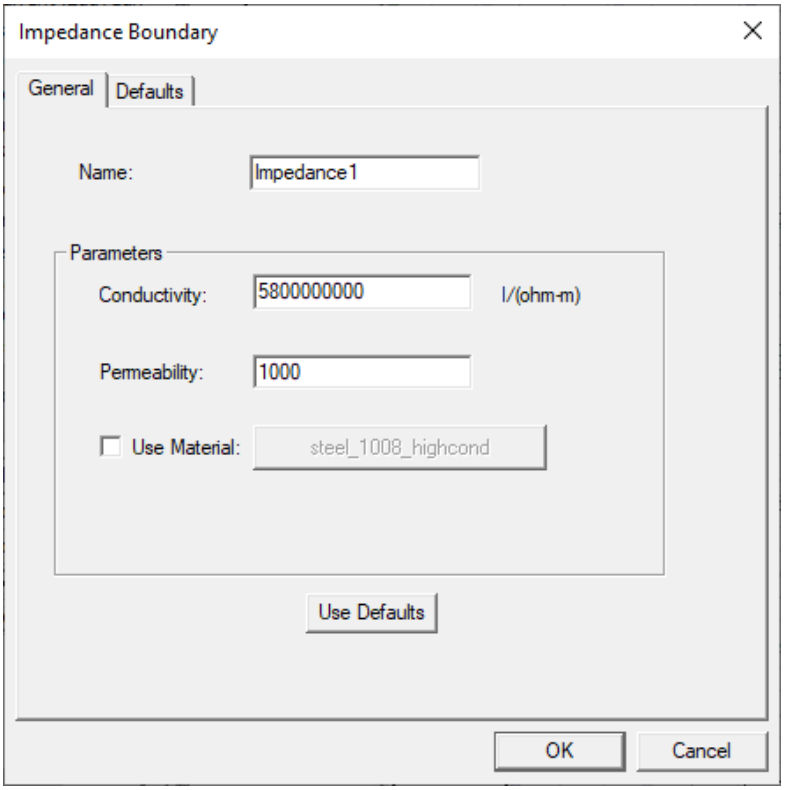
Assigning an Impedance Boundary for the Transient Solver

> **Note:** To define an impedance boundary for the eddy current solver, refer to *Assigning an Impedance Boundary for the Eddy Current Solver*.

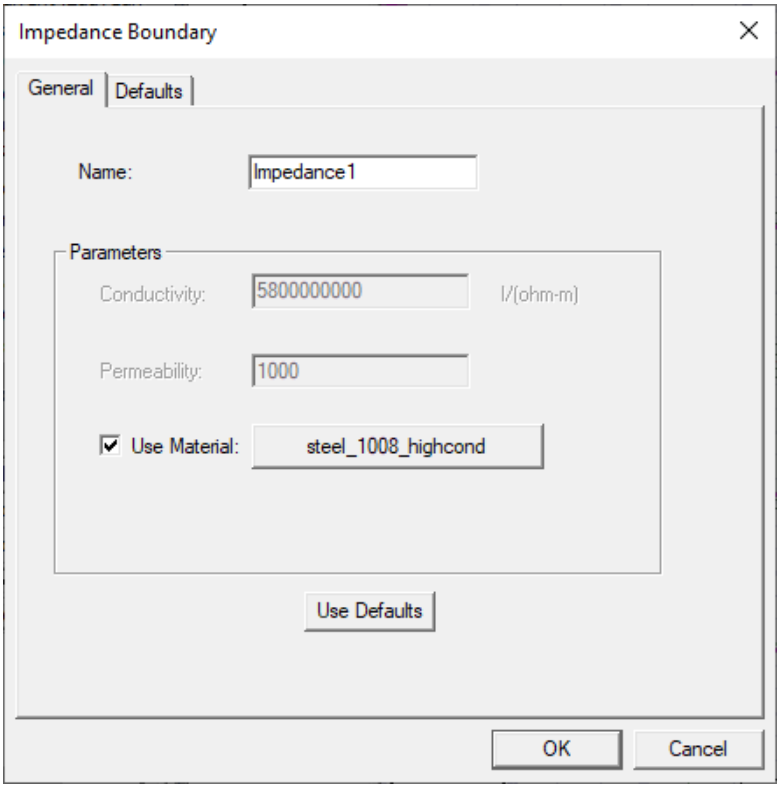
This boundary condition is used to simulate the effect of induced currents in a conductor without explicitly computing them. Since the conductor must be excluded from the model (saving time needed to mesh and solve for currents), assign the impedance boundary condition to an outside edge of the problem region or to an excluded object.

To define an impedance boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).
2. Click **Maxwell 3D>Boundaries>Assign>Impedance** to open the **Impedance Boundary** dialog box.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Do one of the following:
    - If the **Use Material** check box is unchecked, enter the conductivity (in inverse ohm-meters) in the **Conductivity** field, and the conductor's relative permeability in the **Permeability** field.



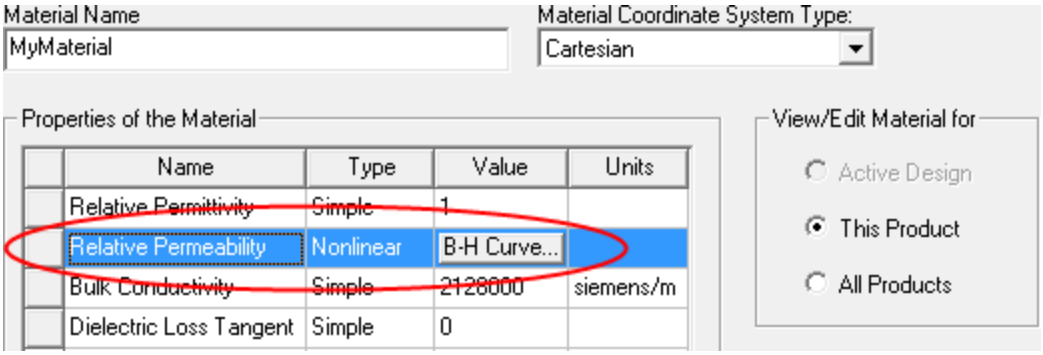
- If you wish to select a material from the library, check the **Use Material** check box, then use the button to select the desired material.



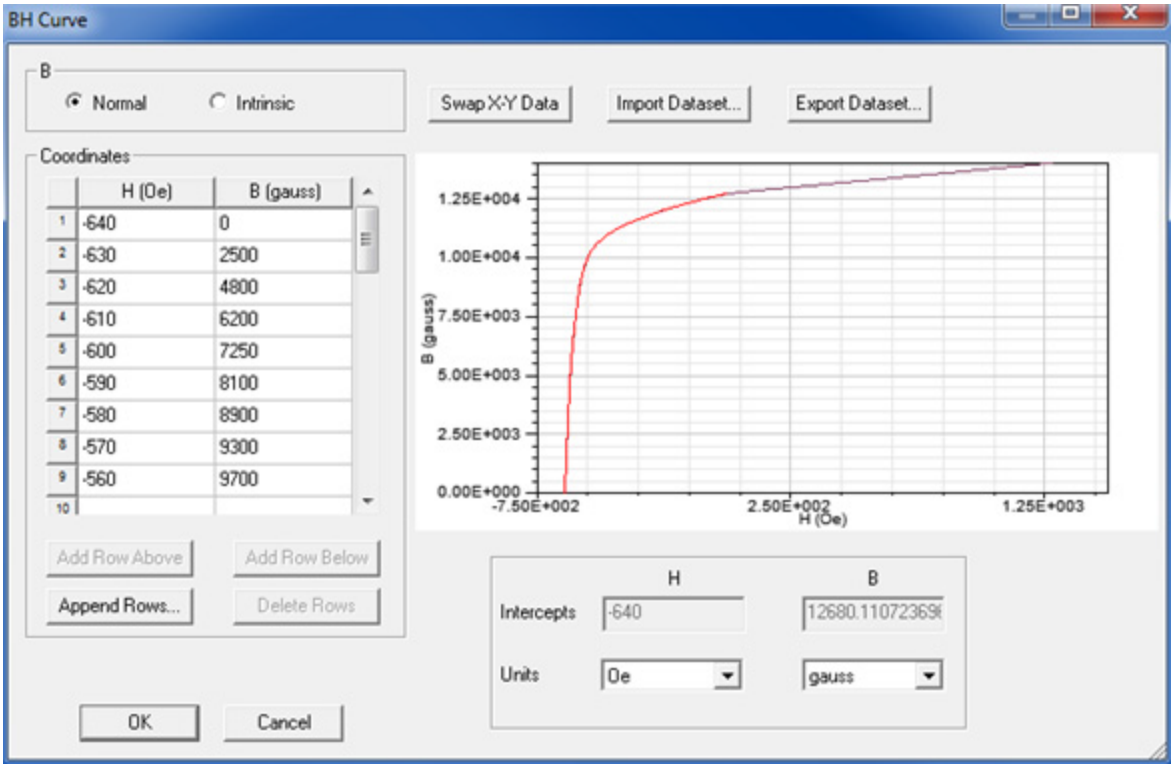
- **Nonlinear Permeability**

  If you select a material that has nonlinear permeability, the non-linear coefficient is computed automatically.

  The nonlinear relative permeability of the material is determined by the B-H curve. If the desired B-H curve of the permeability belongs to an existing material, you can directly use that existing material.



  Otherwise, you can edit a new material with the desired B-H curve.



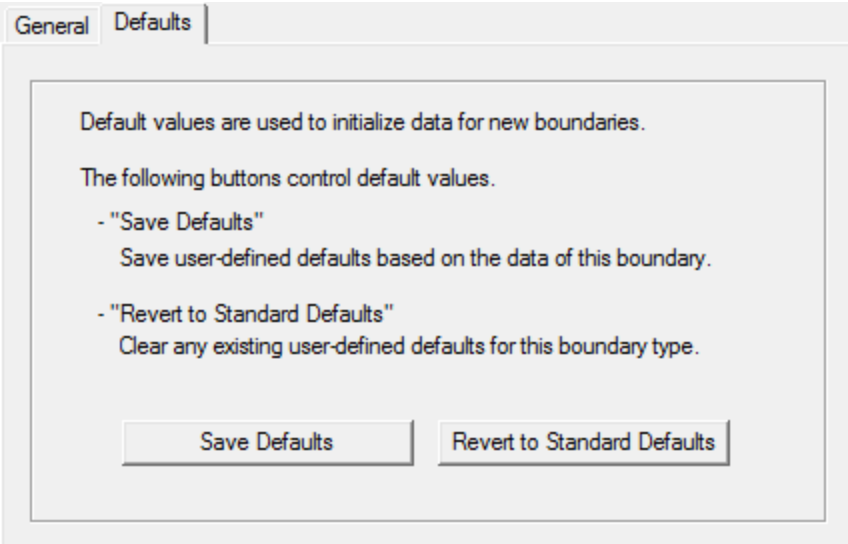
Refer to *Relative Permeability for a Maxwell or RMxprt Material* for details on using nonlinear relative permeability.

- **Temperature Dependent Impedance Boundary**

  The electrical conductivity and permeability of a material used for an impedance boundary condition can be temperature dependent. The Maxwell 3D transient solver provides an option to allow you to input functions as thermal modifiers to consider the effect of temperature on those properties. To enable this feature, check the Use Material check box as above. You can then click Material to open the material properties panel in which you can add a thermal modifier. (Refer to *Setting the Temperature of Objects* for additional information on using temperature dependent impedance boundaries).

5. Optionally, click **Use Defaults** to revert to the default values in the window.

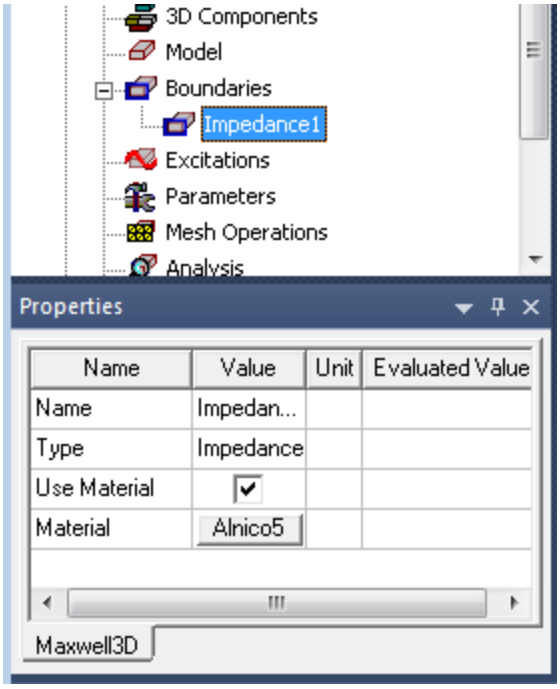
The **Defaults** tab allows you to control default values. The **Save Defaults** button saves the values currently defined on the **General** tab as the defaults to be assigned to new impedance boundaries. **Revert to Standard Defaults** clears existing user-defined values and replaces them with the standard default values.



6. Click **OK** to assign the boundary to the selected object.

   The Project Manager lists the newly assigned impedance boundary in the tree. You can select the boundary in the tree to view and edit its properties in the Properties Window. You

can also double-click the boundary entry in the tree to open it for editing in the Impedance Boundary dialog box.



Related Topics

Impedance Boundary

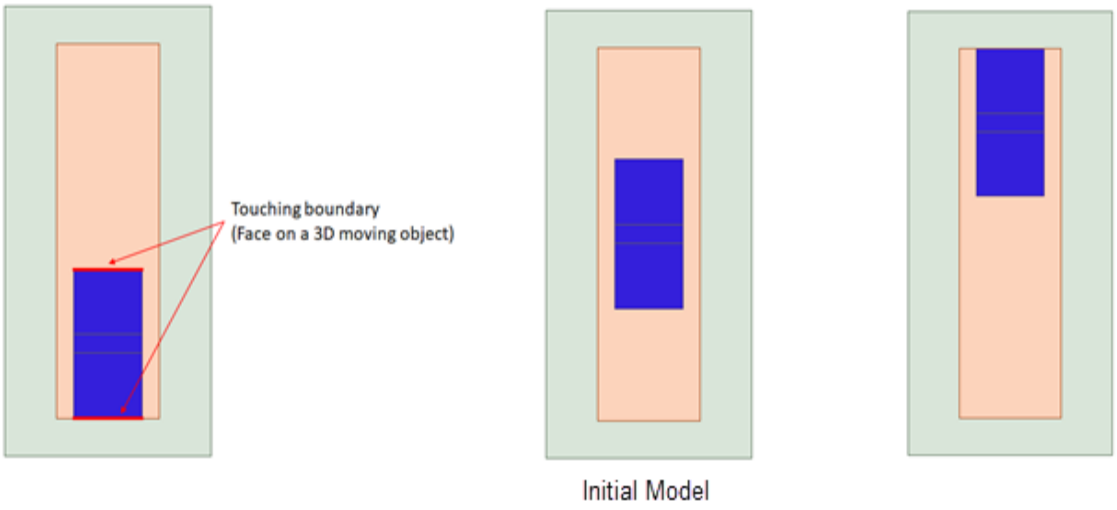
Setting a Thermal Modifier

Setting the Temperature of Objects

Assigning a Touching Boundary for the Transient Solver

For non-periodic translational motion designs in which a moving part contacts a stationary part this boundary condition assigns a touching boundary to the face of the moving part that contacts the stationary part. When the model is created, there must be an airgap between the moving objects and the stationary objects although these can touch during the simulation (either at beginning or end).

For example, with this feature, the linear actuator can be modeled from one end directly touching the stationary part to the other end also directly touching the stationary part as shown; however the initial model must be drawn without touching the ends as shown in the center image below.



To assign a touching boundary to a non-periodic translational motion part:

1. Select the face of the geometry on which you want to apply the boundary condition (the face of the 3D moving body that will touch a stationary part).

Note	• The touching boundary should be defined on the surface of the moving part. • The touching boundary can only be applied to the moving part with non-periodic translation motion. • At any time step, each touching boundary must either fully touch the stationary part or be completely separated from the stationary part. The touching boundary can not partially touch the stationary part at any time. • The moving and stationary parts are not allowed to be on one conduction path. • Moving object(s) cannot touch the stationary objects in the initial model.

2. Click **Maxwell>Boundaries>Assign>Touching**.
   The **Touching Boundary** window appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

Related Topics

Assigning a Band of Motion

Transient Excitations

After defining the coil terminals for the model, you can use these terminals to define one or more windings using the **Maxwell>Excitations>Add Winding** option. Transient problems use the following sources of magnetic fields:

Excitation	Type of Excitation
Coil Terminal	Used to define one or more model windings.
Winding With Current	Current for both a stranded and solid conductor.
Winding With Voltage	Voltage for both a stranded and solid conductor.
Winding With External Circuit Connection	External circuit connection for both a stranded and solid conductor.

Note	A winding is a flexible excitation. Six different combinations can be used: Current, Voltage, and External Circuit — each with a solid or stranded conductor. The Coil Terminal type of excitation is needed to set up a winding.

Note	When a source conductor touches an outer boundary, Maxwell assumes current will flow perpendicular to the boundary. If you do not want current to cross the outer boundary, an insulating boundary should be used.

In addition, permanent magnets serve as sources of magnetic fields.

Assigning a Coil Terminal for a Transient Solver

To assign a coil terminal as a transient excitation:

1. Select the section of the geometry on which you want to apply the excitation (typically a 2D planar object).
2. Click **Maxwell>Excitations>Assign>Coil Terminal**. (You can also right-click an existing winding in the project tree, and select **Assign Coil Terminal**.)
   The **Coil Terminal Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the number of conductors for the coil terminal in the **Number of Conductors** box.

Note	This value represents only the number of conductors inside of the selected geometry. If the Coil Terminal is cut due to symmetry, then only enter the Number of Conductors in the portion modeled.

5. If you want to switch the direction for the coil terminal, click **Swap Direction**.

Note	The direction is shown in the model by a bold red arrow.

6. Optionally, click **Use Defaults** to revert to the default values in the window.
7. Click **OK** to assign the coil terminal to the selected object.

Note	To add this coil terminal to an existing winding, do the following: 1. In the project tree, right-click the coil terminal, and select **Add to Winding**. The **Add to Winding** dialog box appears. 2. Select the winding to which you ant to add the coil terminal, and click **OK**.

Note	To remove a coil terminal from an existing winding, do the following: • In the project tree, right-click the coil terminal, and select **Remove from Winding**. The terminal is removed from the winding and moved up one level in the project tree (directly beneath **Excitations**).

Note	For 3D transient solutions using the A-Phi Formulation, refer to the Winding Excitations for A-Phi Formulation section for information on Ordering the Coil Terminals.

Related Topics

"Assigning a Winding Setup for a Transient Solver " below

Order Coil Terminals for A-Phi Formulation

Assigning a Winding Setup for a Transient Solver

Note	• If you wish to define a winding for an Eddy Current solution type, refer to "Assigning a Winding Setup for an Eddy Current Solver " on page 11-81 . • If you wish to define a winding for a Magnetostatic solution type, refer to Assigning a Winding Setup for a Magnetostatic Solver.

You can use the pre-defined coil terminal(s) to define one or more current or voltage windings.

To define a winding for your model:

1. Click **Maxwell 3D>Excitations>Add Winding**. to open the **Winding** dialog box.
2. Enter a name for the winding in the **Name** box, or accept the default.

3. In the **Parameters** section:
   a. Select **Current, Voltage**, or **External** from the **Type** drop-down menu.

Note	Maxwell transient designs can be dynamically coupled to Twin Builder components through the Twin Builder user interface. You can enable this feature on the Advanced Product Coupling tab of the **Design Settings** dialog box. Source windings set to **External** are available as conservative pins in Twin Builder. For more information, refer to the Twin Builder Help.

   b. Select the **Solid** or **Stranded** radio button to specify the type of conductor.
   c. Enter values in the following fields (enabled according to winding type), and select the desired units.

For a current winding	For a voltage winding[1]	For an external winding
Current	Resistance	Initial Current
	Inductance	
	Voltage	
	Initial Current	

Note	You can also type a function as an expression for any of these fields except Initial Current.

   d. Enter a value in the **Number of parallel branches** text box.

Note	For the **Number of parallel branches** setting, the basic assumption is that the windings are symmetric and balanced, so the current split is forced to be equal in parallel branches. For solid or stranded windings, the **Number of parallel branches** setting assumes there are no circulating currents in the parallel branches. For solid windings, the **Number of parallel branches** setting is intended for applications meeting these conditions: For example, a periodic electric machine wedge model having solid hairpin windings, where the parallel branches are in the symmetric parts of the periodic model. In order to include unbalanced circulating currents this **Number of parallel branches** setting is not applicable, and you must use separate windings for each parallel branch that is connected in an external circuit.

4. Optionally, click **Use Defaults** to revert to the default values in the window.
5. Click **OK** to assign the excitation to the selected object.

A winding is now defined for your model.

Note	To add a coil terminal to this winding, do the following: 1. In the project tree, right-click the winding, and select **Add Terminals**. The **Add Terminals** dialog box appears, listing all coil terminals that do not already belong to that winding. 2. Select the coil terminal (s) you want to add. To select multiple terminals, press

	CTRL and click each terminal. 3. Click **OK**. To assign a coil terminal excitation and add it to this winding, do the following: 1. Select the section of the geometry on which you want to apply the coil terminal excitation. 2. In the project tree, right-click the winding, and select **Assign Coil Terminal**. The **Coil Terminal Excitation** dialog box appears. 3. Enter a name for the excitation in the **Name** box, or accept the default. 4. In the **Parameters** section, enter the number of conductors for the coil in the **Number of Conductor**s box. **Note**: This value represents only the number of conductors inside of the selected geometry. If the Coil Terminal is cut due to symmetry, then only enter the Number of Conductors in the portion modeled. 5. Click **Swap Direction** to change the reference direction for the coil terminal. You can also change this direction in the coil terminal's Properties window by selecting either **Point into terminal**, or **Point out of terminal** for the **Direction** parameter. 6. Optionally, click **Use Defaults** to revert to the default values in the dialog box. 7. Click **OK** to assign the coil terminal excitation. The coil terminal excitation is assigned and is added to the winding. To delete all coil terminal excitations that belong to this winding, do the following: • In the project tree, right-click the winding, and select **Delete All Terminals**. All coil terminal excitations are removed from the winding and deleted from the excitations.

Note	For 3D transient solutions using the A-Phi Formulation, refer to the Winding Excitations for A-Phi Formulation section for information on Ordering the Coil Terminals.

Related Topics

Winding Excitation for A-Phi Formulation

"Assigning a Coil Terminal for a Transient Solver " on page 11-100

"Setting Up an External Circuit Connection for an External Winding Type " on page 11-105

Setting Up a Y Connection in 2D and 3D Transient Designs

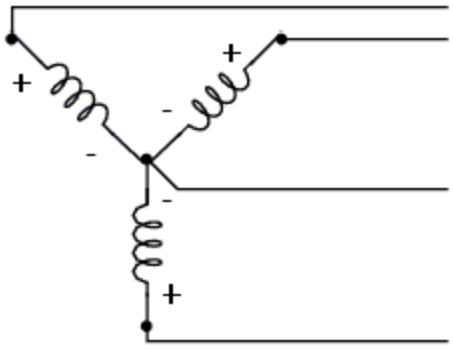
---

1. For a solid winding, the resistance term can represent: the resistance of a portion of the winding which is not modeled (for instance, end-effects), the leads connecting the winding to the source, or the source resistance. (The main winding resistance is calculated directly by the solver.) For a stranded winding, the resistance term is the complete DC resistance of the winding (since the solver does not determine resistance of a stranded winding) as well

as the resistance of the end-effects, leads, source, etc. For both solid and stranded windings, the inductance term can represent: the extra inductance for a portion of the winding of the winding which is not modeled (for instance, end-effects), the leads connecting the winding to the source, or the source inductance. (The main winding inductance itself is calculated directly by the solver.)

Setting Up a Y Connection in 2D and 3D Transient Designs

The Y Connection function available in 2D and 3D Transient solution types allows multiple windings to be connected in a classical Y (sometimes referred to as wye) configuration with the negative terminals connected to a common node as illustrated below.



Setup Y Connection requires voltage winding definitions be set and the negative voltage terminal for each winding will be connected to form the common node of the circuit.

Each voltage source specified in the winding description drives the positive side of the coil. Y Connections are commonly used in motor and transformer applications driven by 3-phase voltage; however, any number of windings may be connected to a common node in Maxwell.

Note	• Only one Y Connection is allowed in a given Maxwell design. • A Y Connection requires use of stranded voltage windings. • Duplicate windings are not allowed.

To define a Y-connection:

1. Click **Maxwell 2D** (or **Maxwell 3D**)**>Excitations>Setup Y Connection**.
   The **Setup Y Connection** dialog appears.
2. In the **Winding** list on the left, select the windings to be grouped into a Y connection.
3. Click the **Group** button to create a Y Connection group with the selected windings.
4. Click **OK** when all required grouping have been created.

Related Topics

Assigning a Winding Setup for a 2D Transient Solver

Assigning a Winding Setup for a 3D Transient Solver

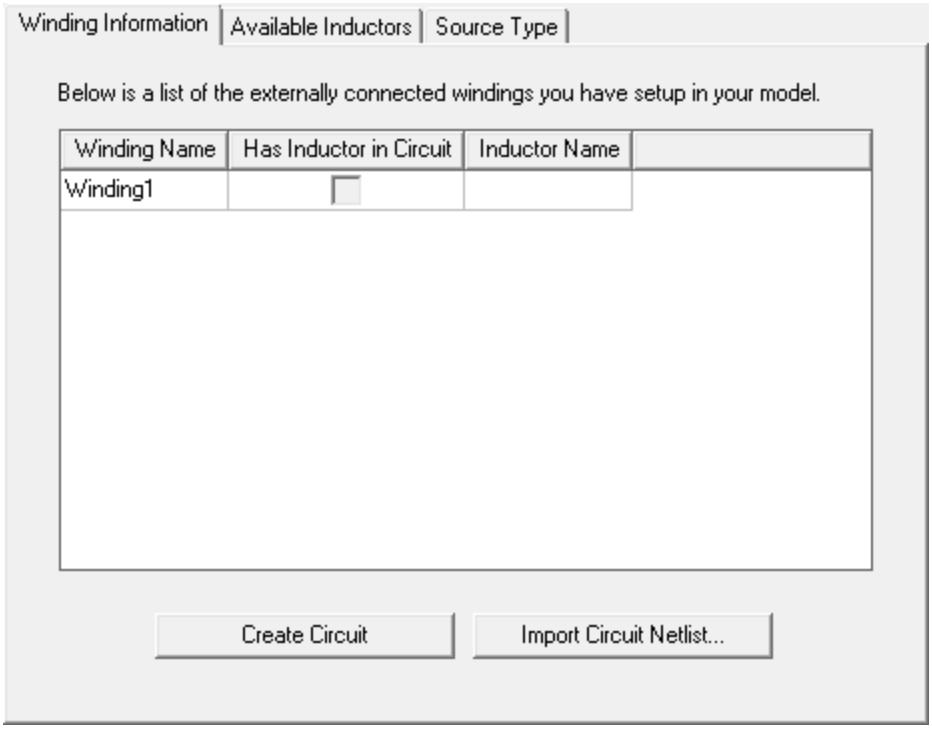
Setting Up an External Circuit Connection for an External Winding Type

The Maxwell Circuit design editor is used to set up the external circuit excitation for windings in Maxwell 2D and 3D transient solutions.

Note	External circuits intended for use with the Time Decomposition Method (TDM) can only support basic circuit elements that are solution-independent. This is because, for TDM, all time steps (or all time steps in a subdivision) are solved simultaneously - and thus the circuit topology and its parameters must be known before solving. Therefore some circuit elements, such as solution-dependent switches, diodes, capacitors, and inductors which are not directly connected to windings in series, are not supported.

To set up an external circuit connection for an External winding type:

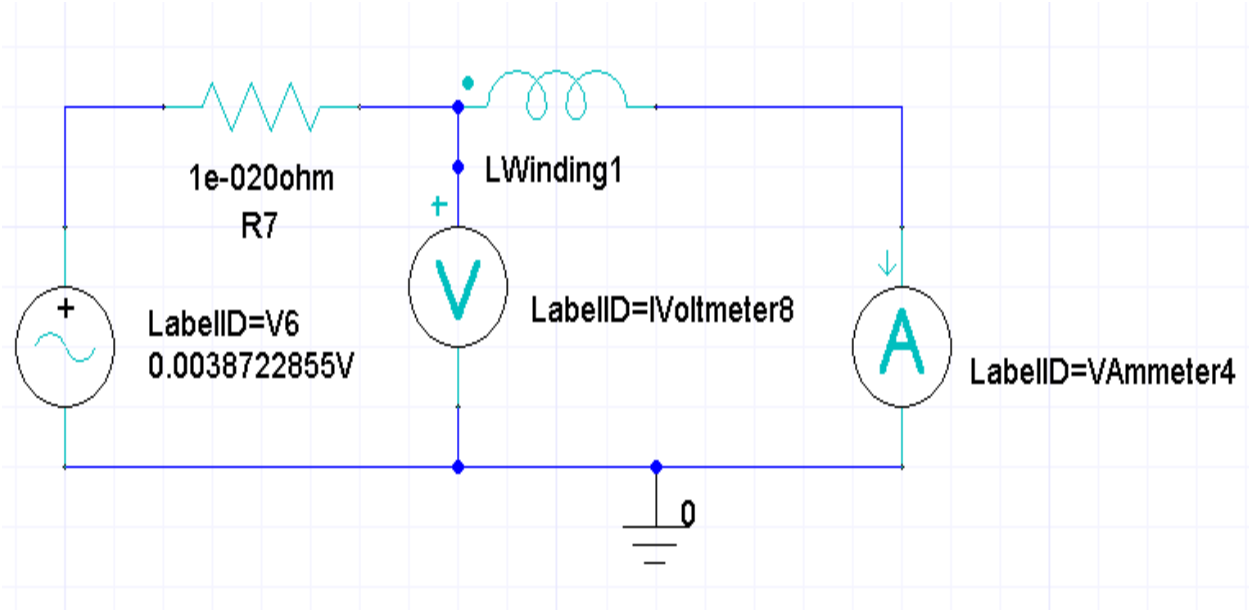
1. Click **Maxwell 3D>Excitations>External Circuit>Edit External Circuit** to open the **Edit External Circuit** dialog box. You can also right-click on **Excitations** in the Project tree and select **External Circuit>Edit External Circuit**.
    a. If no external winding connections have been set up for your model, the **Winding Information** tab lists the external winding(s) in the design. Continue with step 2.



b. If external windings have already been set up, continue with step 7.

2. On the **Winding Information** tab, click **Create Circuit** to add a circuit design to the project. The new circuit design automatically opens for editing in the Maxwell Circuit Editor. The circuit includes one inductor for each external winding in the original Maxwell transient design. The following example is for a design having one external winding.

3. Add the components needed to complete the external circuit design such as shown the following example.



4. When the circuit design is finished, click **Maxwell Circuit>Export Netlist** and export the netlist **.sph** file to the desired location.
5. Return to the Maxwell design and click **Maxwell 3D>Excitations>External Circuit>Edit External Circuit** to open the **Edit External Circuit** dialog box. You can also right-click on **Excitations** in the Project tree and select **External Circuit>Edit External Circuit**.
6. On the **Winding Information** tab, click **Import Circuit** to import the circuit.
7. To view a list of inductors in the imported circuit, click the **Available Inductors** tab.

8. To view the sources used in the externally connected windings, click the **Source Type** tab.
9. If variables are used in the imported circuit, they and their design values are listed on the **Parameter Values** tab. You can link to variables within the imported circuit as follows:
    a. On the **Parameter Values** tab, click in the **Value** column of the parameter to be mapped.
    b. Enter a new local variable name and press the `Enter` key to open the **Add Variable** dialog.
    c. Select a **Unit Type** and **Unit**, and enter a **Value** for the variable and click **OK**. The new Maxwell variable is now mapped to the variable in the imported circuit, and can be varied directly by Maxwell or used for Optimetrics analyses.
10. If the imported file is a .**sph** file, you can click the **Circuit Path** tab to view the original project and design names (which are removed after you import a non-sph file).

Note	After you import a circuit netlist, the **Create Circuit** button changes to **Update Netlist from Circuit**. Use **Update Netlist from Circuit** to reimport the netlist to keep it current.

11. Click **OK** to close the **Edit External Circuit** dialog box.

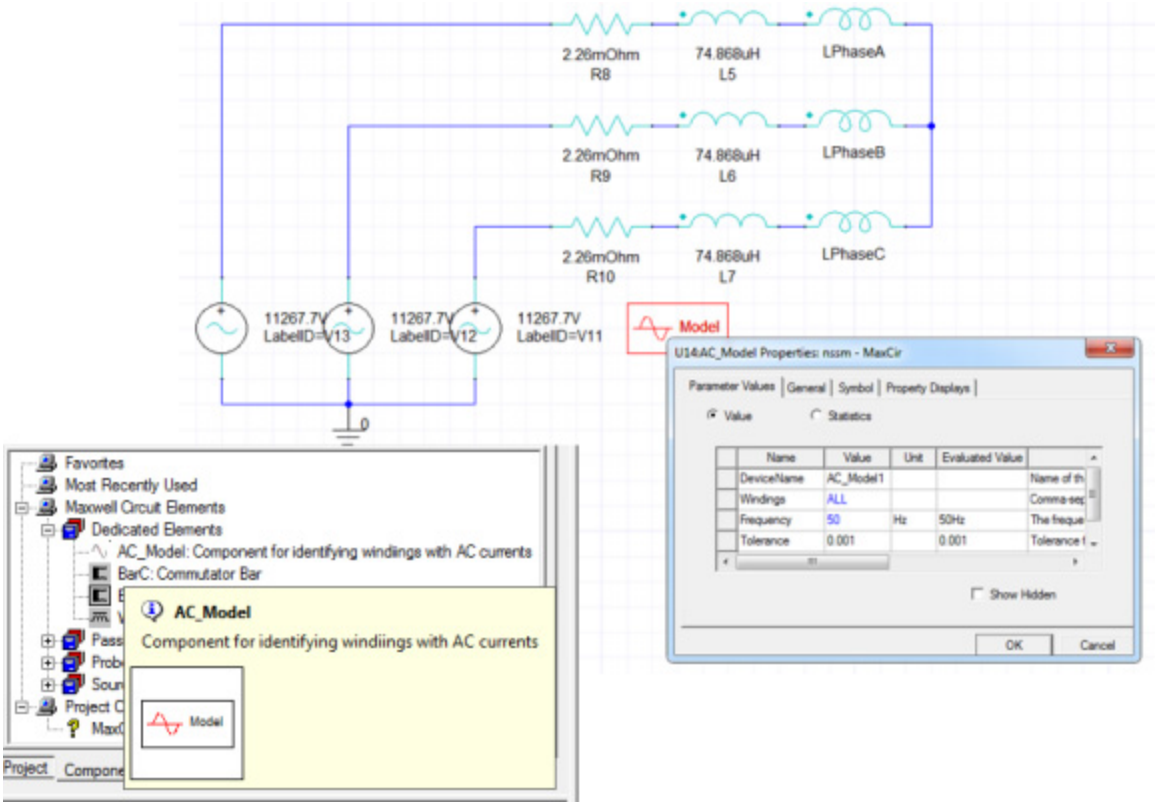
Note	For Transient designs, the user-set time steps can be modified when external circuits are used to drive the windings of a transient finite element model. Following is a list of situations that will lead to a time step change: • All power electronic switching instances that do not coincide with user-specified solve times request a new solution time from the transient solver. The respective switching generating a new solve time request for the transient solver can be time, position, or speed dependent. • When current and/or voltage sources with a piecewise linear variation are used, a new solution time is requested from the transient solver at each (time) definition point used in the corresponding source definition table. • When the change in any winding inductance value is excessive, a new time step (smaller) is calculated and a new solution generated (re-calculated) accordingly. If a user-specified save field time is missed because of the above reasons, the next solved time step fields are saved instead.

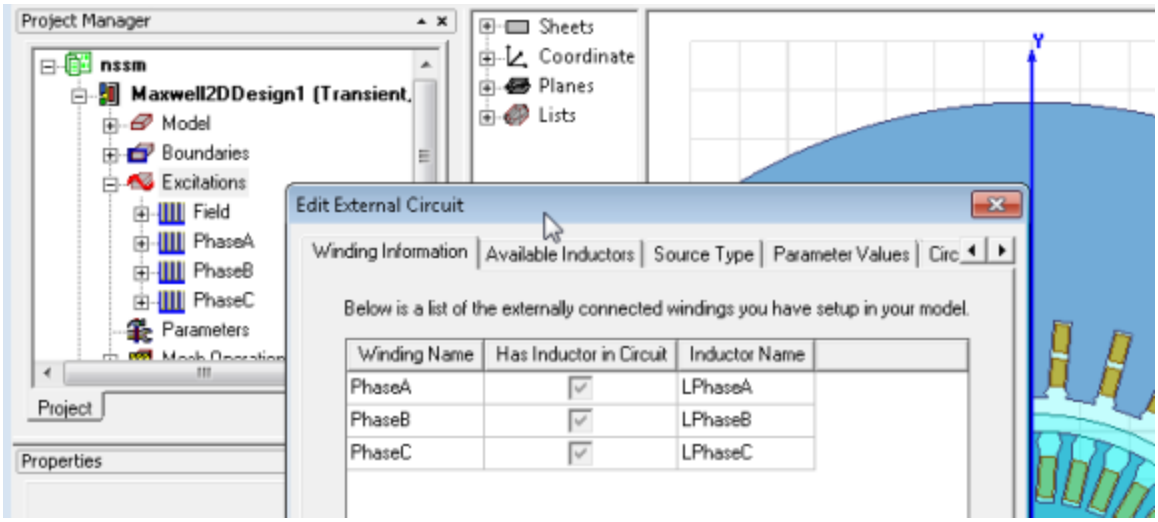
Related Topics

"Assigning a Winding Setup for a Transient Solver " on page 11-101

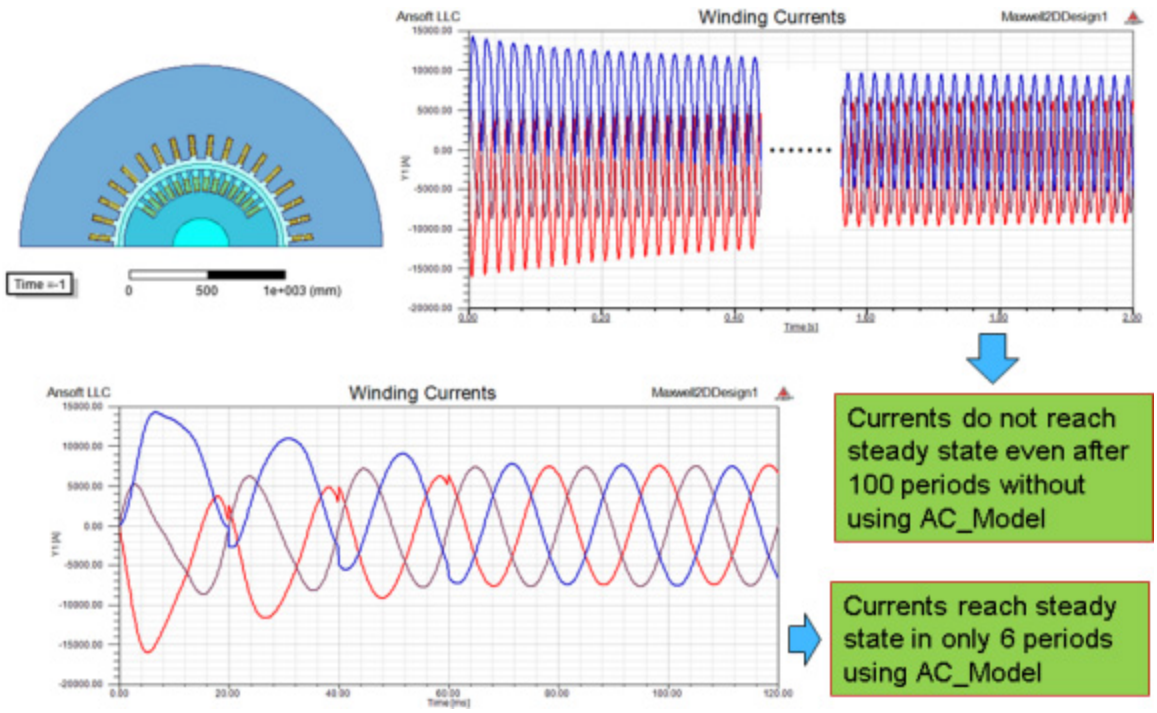
Using the AC_Model Component to Speed Up Analysis

Use of the AC_Model component when setting up an external circuit design can improve transient analysis time to steady state significantly.





In the above example the AC_Model is associated with ALL three of the phase windings. The AC_Model modifies currents for the AC windings at the beginning of each period without interrupting the transient process - thus improving the analysis time.



Setting Up an External Circuit Using Simulink

In addition to using the Maxwell Circuit Editor to set up an external circuit, you also can use Matlab® Simulink® to set up an external circuit for both Maxwell 2D and 3D transient designs.

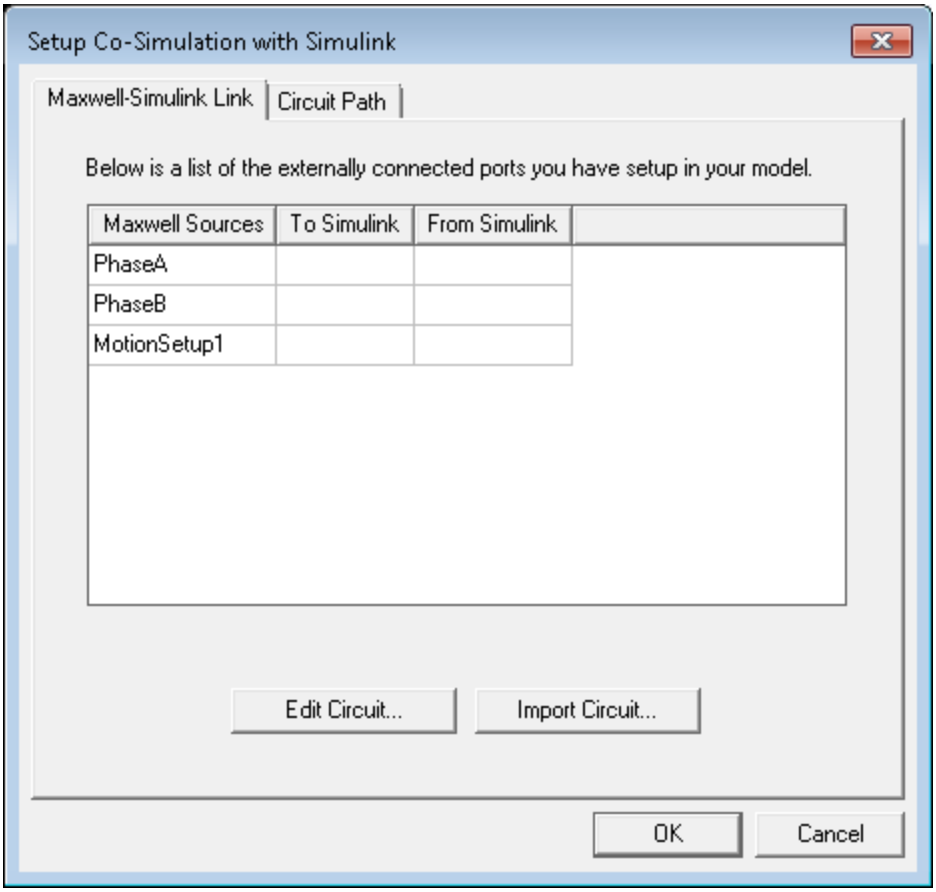
- Supported Simulink versions are: R2017b, R2018a, R2018b, R2019a, R2019b, R2020a, R2020b, and R2021a.

In addition to winding sources, motion sources can also be linked in the external circuit.

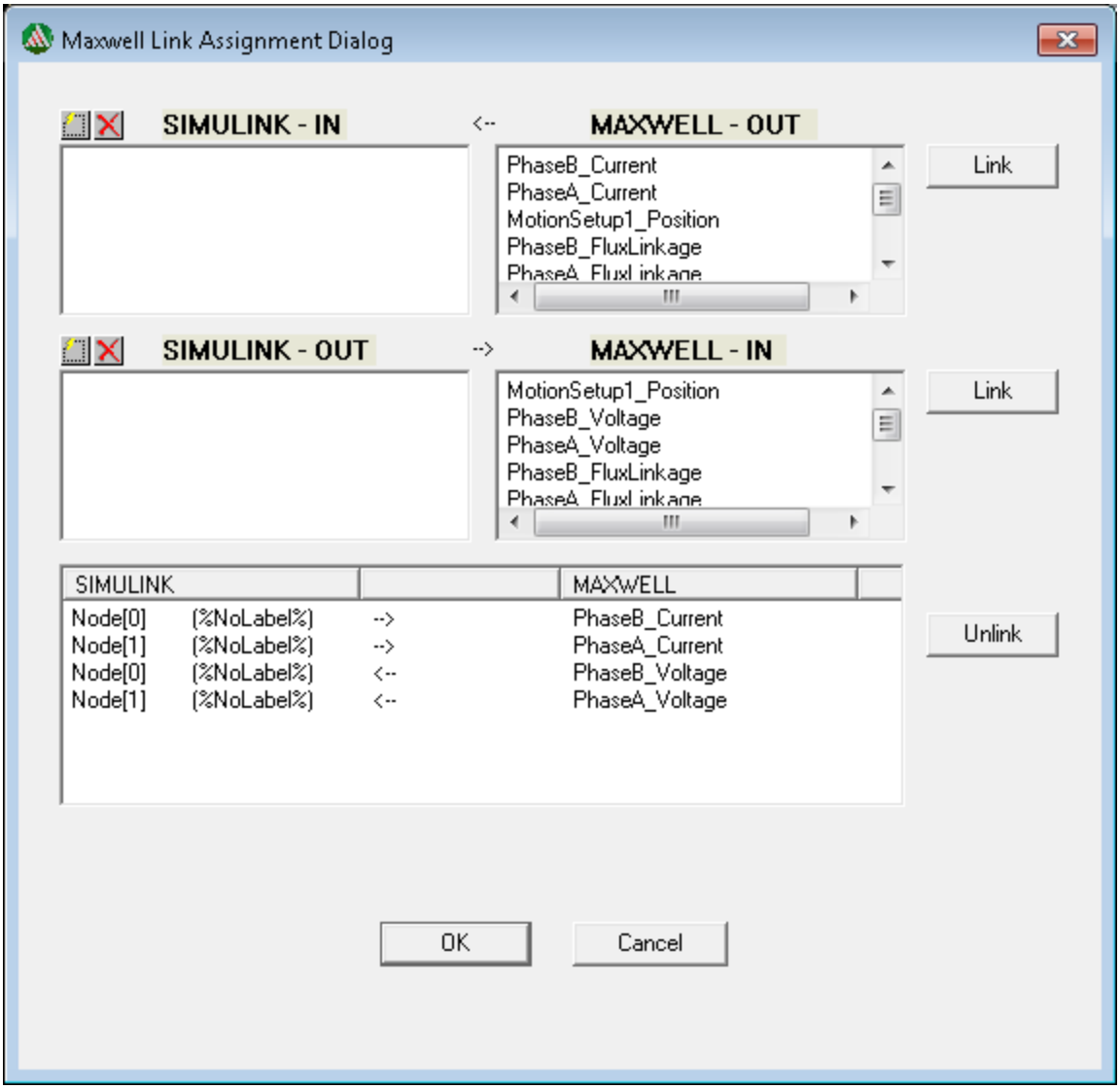
- Winding sources ("External" type only): voltage, current, and flux linkage.
- Motion sources: speed, position, and force/torque

To set up an external circuit using Simulink:

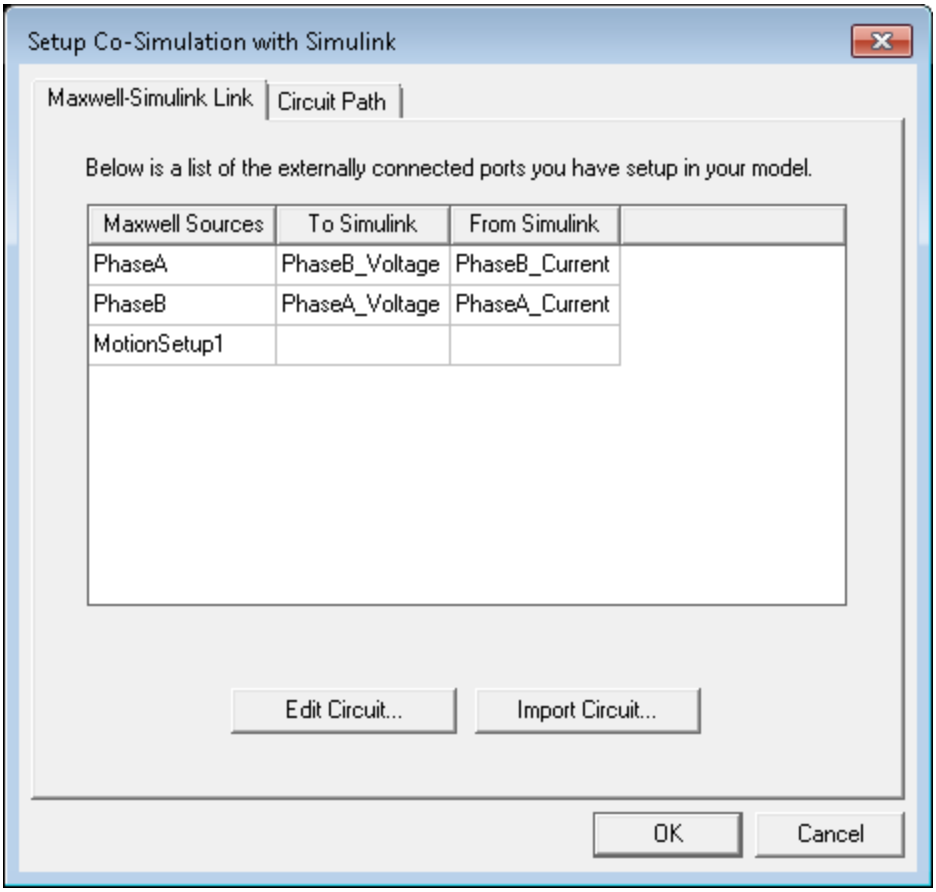
1. In the Maxwell Project Manager, right-click **Excitations** and select **Setup Co-Simulation with Simulink** to open the **Setup Co-Simulation with Simulink** dialog box. You can also open the dialog box from the main **Maxwell 2D** or **3D>Excitations** menu. If an external Simulink circuit has already been created and linked, **To Simulink** and **From Simulink** connections information is shown. Otherwise, these table cells are empty.



2. On the Maxwell-Simulink Link tab, click the **Edit Circuit** button to launch Simulink. The Maxwell sources are listed in the in the **Maxwell Link Assignment Dialog** box. If a circuit has already been created, the circuit also displays in the Simulink circuit editor window.

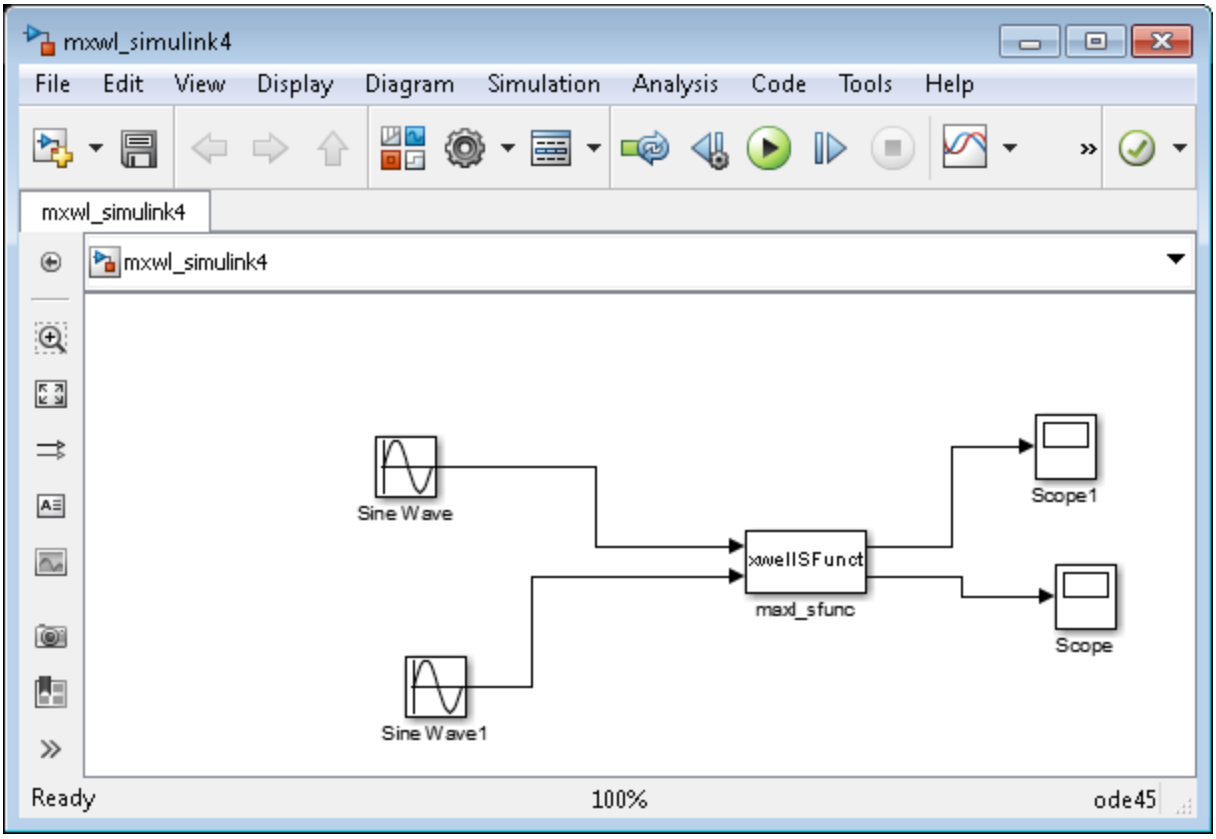


3. Add and **Link** SIMULINK - IN and SIMULINK- OUT connections to the desired Maxwell parameters to set up the Maxwell component ports. You can also **Unlink** connections as needed. Click **OK** once the ports are set up. The ports will show in the Simulink circuit editor window.

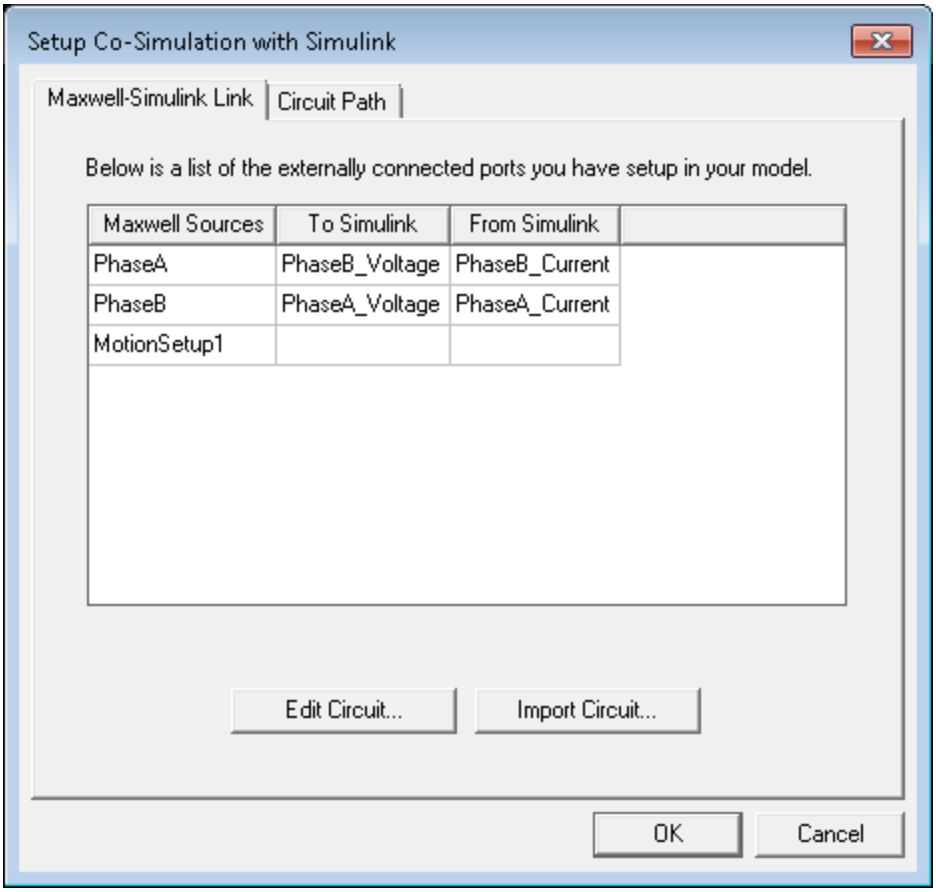


If changes are needed, click **Edit Circuit** to reopen the **Maxwell Link Assignment** dialog box.

4. Modify (or create) the circuit in the Simulink circuit editor window as needed, and save the Simulink circuit project.



5. In the Maxwell **Setup Co-Simulation with Simulink** dialog box, click the **Import Circuit** button to import the Simulink circuit project into the Maxwell project. The linked connections appear in the **Edit External Circuit** dialog box. The **Circuit Path** tab shows the path to the location from which the Simulink circuit file was imported.



Note	You can also modify an existing Simulink circuit by opening the circuit file directly in Simulink, and then importing the modified circuit file back into the Maxwell design.

6. When you start the Maxwell design analysis, the co-simulations are started between Maxwell and Simulink .

Note	• Making changes to the Maxwell winding and motion setups invalidate the external circuit. • Making changes to the external Simulink circuit setup invalidates the solutions.

Related Topics

"Setting Up an External Circuit Using the Maxwell Circuit Editor " on page 27-79

Transient A-Phi Formulation Boundaries and Excitations

Define at least one of the following as a source of magnetic fields:

- A voltage or current.
- Stranded or solid windings, with a voltage or current supply or connected to an external circuit winding.

Note	For stranded conductors, it is assumed that the current density is uniform on the cross-section of the respective conductors. Thus, the stranded option for current excitations should mostly be used only in cases where the cross-section of conductors is constant, consistently with the assumption that the respective object is a coil built with strands of wire. For this reason, conductors (coils) could be created by either sweeping around an axis using zero segments or sweeping along a path having only true surface segments in order to create a smooth surface coil having a completely uniform cross-sectional area over its entire length.

- A permanent magnet.

You may need to set at least one outer boundary to the following:

- The default boundary conditions.
- An odd symmetry (flux tangential) boundary.
- An even symmetry (flux normal) boundary.

Related Topics

Technical Notes: A-Phi Formulation in Maxwell 3D (Transient)

Transient A-Phi Formulation Boundaries

Transient A-Phi Formulation Excitations

Maxwell 3D Transient Solution Based on A-Phi Formulation

Specifying the Solver Type

Transient A-Phi Formulation Boundaries

The transient A-Phi field solver allows you to define the following types of boundary conditions.

> **Note:** The setup for Insulating, Symmetry, Independent, Dependent, and Resistive Sheet boundaries is the same for both Transient and Transient A-Phi Formulation solvers.

Boundary Type	H-Field Behavior	Used to model...
Default Boundary Conditions (Natural and Neumann)	Field behaves as follows: • Natural boundaries –The tangential component of **H** and the normal component of **J** are continuous across the boundary. • Neumann boundaries – **H** is normal	Ordinary field behavior. Initially, object interfaces are natural boundaries; outer boundaries, and excluded objects are Neumann boundaries.

	to the boundary; its tangential components are zero. **J** field is tangential to the boundary.	
Flux Tangential	**H** is tangential to the boundary and flux cannot cross it.	Problems with flux tangential at outer boundary.
Insulating	Same as Natural boundary, except that current cannot cross the boundary.	Thin, perfectly insulating sheets between touching conductors.
Symmetry	Field behaves as follows: • Odd Symmetry (Flux Tangential) – **H** is tangential to the boundary; its normal components are zero. The potential value at the boundary is zero. • Even Symmetry (Flux Normal) – **H** is normal to the boundary; its tangential components are zero.	Planes of geometric and magnetic symmetry.
Matching (Independent and Dependent)	The **H**-field on the dependent boundary is forced to match the magnitude and direction (or the negative of the direction) of the **H**-field on the independent boundary.	Planes of symmetry in periodic structures where **H** is oblique to the boundary.
Resistive Sheet	Same as Natural boundary.	Voltage drop and loss on the contact surface between two different objects within a conduction path.

Default Boundary Conditions for a Transient A-Phi Solver

These boundary conditions are automatically defined for a transient model:

- Natural boundaries are assigned to the surfaces between objects.
- Neumann boundaries are assigned to the outer surface of the problem region. **J** field is tangential to the boundary, and **H** field is normal to the boundary.

To leave a surface set to its default boundary condition, you do not need to set any boundary conditions. Deleted boundary conditions and excitations automatically reset to the default boundary conditions.

Assigning a Flux Tangential Boundary for a Transient A-Phi Solver

This type of boundary is applied when the **H** field is tangential to the boundary.

To define a zero tangential **H** field boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically a face).

2. Click **Maxwell>Boundaries>Assign>Flux Tangential**.
3. The **Flux Tangential** window appears.
4. Enter a name for the boundary in the **Name** box, or accept the default.
5. Click **OK** to assign the boundary to the selected object.

Transient A-Phi Formulation Excitations

Transient A-Phi formulation supports the following excitations:

Excitation	Type of Excitation
Coil Terminal	Used to define windings.
Winding With Current	Current for a stranded or solid conductor.
Winding With Voltage	Voltage for a stranded or solid conductor.
Winding With External Circuit Connection	External circuit connection for a stranded or solid conductor.
Voltage	Voltage for a solid conductor.
Current	Current for a stranded or solid conductor.
Permanent Magnet Field	Permanent magnets serve as sources of magnetic fields.

Note	A winding is a flexible excitation. Six different combinations can be used: Current, Voltage, and External Circuit — each with a solid or stranded conductor. The Coil Terminal type of excitation is needed to set up a winding.

Note	Winding excitation is assigned as in the transient solver. However, coil terminals must be ordered according to the physical potential distribution as explained in Boundaries and Excitations>Winding Excitations for A-Phi Formulation.

Electric Transient Solver Boundaries and Excitations

Specify at least one of the following excitations as a source of electric fields:

- The total charge on object.
- The volume charge density inside an object.
- The current.
- Define the electric potential on each surface using a voltage excitation.

Related Topics

Electric Transient Excitations

Specifying the Solver Type

Electric Transient Solver

Electric Transient Excitations

The following excitations are available for electric transient problems:

- Voltage - this excitation specifies the value of the electric scalar potential value on 2D surfaces or 3D volume bodies.
- Charge - this excitation can be applied to 2D surfaces or 3D volume bodies, if a body has net charge q. This type of excitation defines the total charge on a surface or object.
- Floating - this excitation can be applied to 2D surfaces or 3D volume bodies, then these 2d or 3d objects are with equi-potential. This type of excitation models conductors at unknown potentials and specifies the total charge on the conductor.
- Volume Charge Density - this excitation can be applied to only 3D volume bodies. If that body has net charge q, the charge density is q/VOL, where VOL stand for the volume of that body.
- Current - this excitation can be applied to only 2D surfaces, where the total current flowing into/out that surface is specified.
- Sink - this excitation can be applied to only 2D surfaces which are grounded with reference potential value. It is the same as applying zero voltage excitation on those surfaces.

Related Topics

Specifying the Solver Type
Electric Transient Solver

Assigning a Voltage Excitation for an Electric Transient Solver

To define a voltage excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell>Excitations>Assign>Voltage**.
   The **Voltage Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the voltage in the **Value** box, and select the units. Then, select a **Coordinate System**.
5. Optionally, click **Use Defaults** to revert to the default values.
6. Click **OK** to assign the excitation to the selected object.

Related Topics

Electric Transient Excitations
Specifying the Solver Type
Electric Transient Solver

Assigning a Charge Excitation for an Electric Transient Solver

To define a charge excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell>Excitations>Assign>Charge**.
   The **Charge Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the charge in the **Value** box.
5. Optionally, click **Use Defaults** to revert to the default values.
6. Click **OK** to assign the excitation to the selected object.

Related Topics

Electric Transient Excitations
Specifying the Solver Type
Electric Transient Solver

Assigning a Floating Excitation for an Electric Transient Solver

To define a floating excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell>Excitations>Assign>Floating** .
   The **Floating Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the floating excitation in the **Value** box.
5. Optionally, click **Use Defaults** to revert to the default values.
6. Click **OK** to assign the excitation to the selected object.

Related Topics

Electric Transient Excitations
Specifying the Solver Type
Electric Transient Solver

Assigning a Volume Charge Density Excitation for an Electric Transient Solver

To define a volume charge density excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell>Excitations>Assign>Volume Charge Density**.
   The **Volume Charge Density Excitation** dialog box appears.

3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the volume charge density in the **Value** box. Then, select a **Coordinate System**.
5. Optionally, click **Use Defaults** to revert to the default values.
6. Click **OK** to assign the excitation to the selected object.

Related Topics

Electric Transient Excitations
Specifying the Solver Type
Electric Transient Solver

Assigning a Current Excitation for an Electric Transient Solver

To define a current excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell>Excitations>Assign>Current**.
   The **Current Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the current excitation in the **Value** box, and select the units.
5. To swap the direction of the current, click **Swap Direction**.
6. Optionally, click **Use Defaults** to revert to the default values.
7. Click **OK** to assign the excitation to the selected object.

Related Topics

Electric Transient Excitations
Specifying the Solver Type
Electric Transient Solver

Assigning a Sink Excitation for an Electric Transient Solver

To define a current excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell>Excitations>Assign>Sink**.
   The **Sink Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. Click **OK** to assign the excitation to the selected object.

Related Topics

Electric Transient Excitations
Specifying the Solver Type
Electric Transient Solver

12 - Assigning Boundaries and Excitations for 2D Designs

For every design, you need to assign boundaries and excitations. You can use the **Maxwell 2D** menu or the project tree to do the following:

- Define boundary conditions that control how the electric or magnetic field behaves planes of symmetry, periodicity, or edges of the problem region.
- Define solution type specific excitations of voltage, charge, coil, and current.
- For magnetic transient designs, if needed, set up a winding and an external circuit connection.

Each field solver requires you to specify excitations of electric or magnetic fields and references for computing these fields.

- Magnetostatic Boundaries and Excitations
- Electrostatic Boundaries and Excitations
- AC Conduction Boundaries and Excitation
- DC Conduction Boundaries and Excitations
- Eddy Current Boundaries and Excitations
- Transient Boundaries and Excitations

You must specify at least one of the boundary conditions or excitations listed in these sections so that the simulator can compute accurate values for fields and parameters.

Related Topics

Specifying the Solver Type

Selecting Objects and Faces in the 2D Geometry

Before creating a boundary or excitation, you must specify its location by selecting the object or surface on which to assign the boundary condition or source excitation.

Defining Boundary Conditions in 2D

To assign a boundary condition:

1. Select the section of the geometry on which you want to apply the boundary condition.
2. Click **Maxwell 2D>Boundaries>Assign**, or right-click **Boundaries** in the project tree and click **Assign**.
3. Select the desired boundary type from the sub-menu. The specific boundary types you can define depend on the solver you have chosen for your project.
    - Magnetostatic Boundary Conditions
    - Electrostatic Boundary Conditions

- AC Conduction (Electric) Boundary Conditions
- DC Conduction (Electric) Boundary Conditions
- Eddy Current Boundary Conditions
- 2D Transient Boundary Conditions

4. Enter the name for the boundary or accept the default.
5. Depending on the boundary type, you may need to define additional parameters.
6. Click **OK** to create the new boundary.

The new boundary is added to the boundary list in the project tree.

Defining 2D Excitations

The specific excitations you can define depend on the solver you have chosen for your design.

- Magnetostatic Excitations
- Electrostatic (Electric) Excitations
- AC Conduction (Electric) Excitations
- DC Conduction (Electric) Excitations
- Eddy Current Excitations
- 2D Transient Excitations

To assign an excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell 2D>Excitations>Assign**, or right-click **Excitations** in the project tree and click **Assign**.

   A submenu appears with choices that depend upon the solver type.
3. Select one of the following excitation types from the submenu:
   - Voltage
   - Charge
   - Floating
   - Charge Density
   - Current Density
   - Current
   - Coil
   - End Connection
   - Permanent Magnet Field
4. Enter the name for the excitation type or accept the default.
5. Enter the desired parameter values, units, and other settings for the excitation in the dialog box. You can also enter a mathematical function to determine values for voltage, current density, charge density, and other parameters. The Permanent Magnet Field excitation also requires you set up a link to a source project.

Note	When entering current, or coil terminals, the arrow associated with current in the

	selected object shows the direction as if a positive value is entered for the current. If a negative value is entered the actual current flow direction is opposite to what the arrow shows. Click **Swap Direction** if you want to reverse the direction of the arrow.

6. Click **OK** to create the new excitation.

The new excitation is added to the excitation list in the project tree.

Related Topics

Specifying the Solver Type

Viewing 2D Boundaries and Excitations

To view a list of all boundaries or excitations set for the design:

1. In the project tree, right-click **Boundaries** or **Excitations**.
   A shortcut menu appears.
2. Select **List** from the shortcut menu.
   The **Design List** dialog box appears, with the tab page visible for the item you selected. This dialog box contains tabs for **Model**, **Boundaries**, **Excitations**, **Parameters**, **Mesh Operations**, and **Analysis Setup**.
3. To view the properties for an item in the list, select the item, and click **Properties**.
4. To delete an item from the list, select the item, and click **Delete**.
5. When you are finished in the **Design List** dialog box, click **Done** to close it.

Setting the Visibility for 2D Boundaries and Excitations

To show or hide a boundary:

1. Click **View>Active View Visibility**.
   The **Active View Visibility** dialog box appears.
2. Click the **Boundaries** tab.
3. Select or clear the **Visibility** check box for each boundary listed.
4. Click **Done**.

To show or hide a excitation:

1. Click **View>Active View Visibility**.
   The **Active View Visibility** dialog box appears.
2. Click the **Excitations** tab.
3. Select or clear the **Visibility** check box for each excitation listed.
4. Click **Done**.

Functional Boundaries and Excitations in 2D

Functional boundaries and excitations can be used to do the following:

- Define the value of a boundary or excitation quantity (such as the voltage, magnetic field, or current density) using a mathematical relationship — such as one relating its value to that of another quantity.
- Define the value of a boundary or excitation as a function of position.
  - Current Density Excitation (Magnetostatic or Eddy Current solver type)
  - Voltage Excitation (Electric solver types)
  - Volume Charge Density Excitation (Electrostatic or Electrostatic + DC solver types)
- If the parametric analysis capability is installed, identify which boundary or excitation quantities are to be varied during a parametric sweep. These variables are always set to constant values in the parametric analysis.

Setting Eddy Effects and Displacement Current in 2D

(2D Eddy Current and/or Transient Solvers)

For both eddy current and transient solvers in 2D, you may need to specify the following additional settings:

- The behavior of eddy currents and the AC magnetic field in conductors. When you activate the **Eddy Effect** setting, the solver computes the induced eddy currents.
- For 2D Transient, source settings override the eddy effect settings. Objects assigned as “solid” sources will have eddy effects turned on, while objects assigned as “stranded” sources will have eddy effects turned off (regardless of the eddy effect settings). The eddy effect settings will be respected for non-sources.

Typically, background objects are excluded from eddy current settings.

To set eddy effects:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell2D>Excitations>Set Eddy Effects**.
   The **Set Eddy Effect** dialog box appears showing only objects which are valid for eddy effect calculation, specifically objects with conductivity greater than 1.
3. For each object in the list, select or clear the check boxes for **Eddy Effect**. When selecting multiple entries, the changes are synchronized for all selected rows.

Note	Multi-select is supported by holding the **CTRL** or **Shift** key while clicking.

4. Click the **Select By Name** button to enter the name of a specific object for selection. Multiple objects may be selected using the * and ? wildcards.
5. Optionally, click the **Use suggested values** button to set the **Eddy Effects** for all elements recommended by Maxwell2D.
6. Click **OK**.

The **Deselect All** button will deselect all selected objects in the list.

Related Topics

Technical Notes: *Eddy Current Field Simulation*
Technical Notes: *Transient Simulation*

Modifying 2D Boundary Conditions and Excitations

Save your project before modifying boundary conditions and excitations. You can modify boundary conditions and excitations after a solution has been generated, but you may lose some solution data.

Related Topics

Duplicating 2D Boundaries and Excitations
Setting Default Values for 2D Boundaries and Excitations
Reassigning 2D Boundaries
Reassigning 2D Excitations

Duplicating 2D Boundaries and Excitations

To duplicate a boundary or excitation when its geometry is pasted or duplicated:

1. Click **Tools**>**Options**>**Maxwell 2D Options**.
   The **Maxwell 2D Options** dialog box appears.
2. Click the **General Options** tab.
3. Select **Duplicate boundaries with geometry**.
4. Click **OK**.
   All boundaries and excitations are duplicated with their associated geometries until you clear this option.

Hint	Use this option to copy and paste boundaries. For example, do the following to use the same boundary on multiple objects: 1. Select the face to which you want to assign the boundary. 2. Click **Modeler**>**Surface**>**Create Object From Face** to create a new object from the existing face. 3. Assign the boundary to the new face object. 4. Copy and paste the new face object to copy and paste the boundary.

Related Topics

Copying and Pasting Objects

Showing and Hiding Boundaries and Excitations in 2D

You can choose to show or hide a boundary or excitation's geometry, name, or vectors, in the active view window or in all view windows.

Related Topics

Showing and Hiding 2D Boundaries/Excitations in Every View Window

Showing and Hiding 2D Boundaries and Excitations in Every View Window

1. Click **Maxwell 2D>Boundaries>Visualization** or **Maxwell 2D>Excitations>Visualization**.
   The **Boundary Visualization Options** dialog box appears.
2. Select one of the following from the **Select** pull-down menu:
   - All
   - All Boundaries
   - All Excitations
   - By Name
3. If you select **By Name**, enter a name in the **Select By Name** dialog box, and click **OK**.
4. To clear the selection of all boundaries and excitations, click **Deselect All**.
5. Click **Close** to close the **Boundary Visualization Options** dialog box.

Setting Default Values for 2D Boundaries and Excitations

When assigning a boundary or excitation, many of the fields in the boundary and excitation dialog boxes have default values associated with them. These default values are initially set by **Maxwell 2D**, but can be overridden.

To modify the default values associated with a specific boundary or excitation type:

1. Assign a boundary or excitation.
2. Modify any default values.
3. Close the boundary or excitation's dialog box.
4. Re-open the new boundary or excitation's dialog box by right-clicking the boundary or excitation in the project manager and selecting **Properties**. The dialog now includes a **Defaults** tab.
5. Under the **Defaults** tab, click **Save Defaults**.

   The values assigned to this boundary are saved as the default values and are assigned when new boundaries of this type are created.
6. Optionally, click **Revert to Standard Defaults**.

   The default values you set for this boundary type are cleared and will revert to the default values set by **Maxwell 2D**.

Reassigning 2D Boundaries

You can reassign a boundary to another surface. This is useful when you modify objects with assigned boundaries, invalidating the boundaries. For example, if you unite two objects with assigned boundaries, the second object's boundary becomes invalid because united objects maintain the characteristics of the first object selected. In this case, you would need to reassign the boundary or delete it.

To reassign a boundary:

1. Select the object or object face to which you want to assign an existing boundary.
2. Click **Maxwell 2D>Boundaries>Reassign**.
   The **Reassign Boundary** dialog box appears.

Hint	Alternatively, select the object or object face to which you want to assign an existing boundary. Right-click the existing boundary in the project tree, and then click **Reassign** on the shortcut menu.

3. Select the boundary you want to reassign, and click **OK**.

Note	When reassigning a boundary that includes vectors in its definition, Maxwell attempts to preserve the vectors with the new assignment, but this is not always possible.

Related Topics

Reassigning 2D Excitations

Reassigning 2D Excitations

To reassign an excitation:

1. Select the object or object face to which you want to assign an existing excitation.
2. Click **Maxwell 2D>Excitations>Reassign**.
   The **Reassign Excitation** dialog box appears.

Hint	Alternatively, select the object or object face to which you want to assign an existing excitation. Right-click the existing excitation in the project tree, and then click **Reassign** on the shortcut menu.

3. Select the excitation you want to reassign, and click **OK**.

Note	When reassigning an excitation that includes vectors in its definition, Maxwell attempts to preserve the vectors with the new assignment, but this is not always possible.

Related Topics

Reassigning 2D Boundaries

Deleting All 2D Boundaries and/or Excitations

To delete all boundaries or excitations:

- Click **Maxwell 2D>Boundaries>Delete All** or **Maxwell 2D>Excitations>Delete All**.

Viewing and Editing 2D Boundary or Excitation Properties

To open the **Properties** dialog box for a boundary or excitation that is already assigned:

1. In the project tree, under either **Boundaries** or **Excitations**, right-click a specific boundary or excitation, and select **Properties**.
   The **Properties** dialog box for that boundary/excitation appears.
2. Make the desired changes, and click **OK**.

Related Topics

Assigning Boundaries and Excitations for 2D Designs
Modifying 2D Boundary Conditions and Excitations

2D Magnetostatic Boundaries and Excitations

Define at least one of the following as a source of static magnetic fields:

- The current in a conduction path, which can be either stranded or solid.
- The current density in a conductor.
- The magnetic field on an outside surface.

Related Topics

2D Magnetostatic Boundaries
2D Magnetostatic Excitations
Technical Notes: Magnetostatic Field Calculation
Specifying the Solver Type
Permanent Magnet Excitations

2D Magnetostatic Boundaries

The magnetostatic field solver allows you to define the following types of boundaries:

Boundary Type	H-Field Behavior	Used to model...
Default Boundary	Field behaves as follows: • At the interface between objects,	Initially, object interfaces are natural boundaries; outer boundaries, and

Conditions (Natural and Neumann)	the H tangent and B normal are continuous. • On outer boundaries, B field is normal. As a rule, this behavior needs to be modified at least for some portions of the outer boundary to insure uniqueness of the solution.	excluded objects are Neumann boundaries.
Vector Potential	Sets the magnetic vector potential A_z, or rA_ϕ, on the boundary. The behavior of H depends on whether A_z or rA_ϕ is constant or functional.	Outer boundaries at specific vector potentials; externally applied magnetic fields.
Balloon	This boundary is considered to be far away from the other sources in the problem.	Actual behavior is context dependent. In general, if a Dirichlet BC is not applied elsewhere to insure uniqueness, the "balloon" will be assigned a Dirichlet boundary condition.
Symmetry	Field behaves as follows: • Odd Symmetry (Flux Tangential) — **Magnetic Field** is tangential to the boundary; its normal components are zero. • Even Symmetry (Flux Normal) — **Magnetic Field** is normal to the boundary; its tangential components are zero.	Planes of geometric and magnetic symmetry.
Matching (Independent and Dependent)	The **Magnetic Field vector** on the dependent boundary is forced to match the magnitude and direction (or the negative of the direction) of the **Magnetic Field** vector on the independent boundary.	Planes of symmetry in periodic structures where **Magnetic Field** is oblique to the boundary.

Default Boundary Conditions for a 2D Magnetostatic Solver

These boundary conditions are automatically defined for a magnetostatic model:

- Natural boundaries are assigned to the surfaces between objects. At the interface between objects, the H tangent and B normal are continuous.

- Neumann boundaries are assigned to the outside edges of the problem region. On outer boundaries, B field is normal. As a rule, this behavior needs to be modified at least for some portions of the outer boundary to insure uniqueness of the solution.

To leave a surface set to its default boundary condition, do nothing. Deleted boundary conditions and excitations automatically reset to the default boundary conditions.

Assigning a Vector Potential for a 2D Magnetostatic Solver

To define a vector potential boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge). **Maxwell2D** requires that the Vector Potential be assigned on a 1D object such as an edge or a line.
2. Click **Maxwell2D>Boundaries>Assign>Vector Potential**.
   The **Vector Potential Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the value of the potential. The value may be specified as a single numeric value or a functional value.
5. If a functional value is specified, select a **Coordinate System** from the pull-down list:
6. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
7. Click **OK** to assign the boundary to the selected object.

The new boundary is added to the boundary list in the project tree.

Assigning a Symmetry Boundary Condition for a 2D Magnetostatic Solver

This boundary condition defines a plane of geometric or magnetic symmetry in a structure. Assign it only to the outer surfaces of the problem region.

To set a symmetry boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Symmetry**.
   The **Symmetry Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select one of the following as the type of symmetry:

Odd (Flux Tangential)	**Magnetic Field** is tangential to the boundary; its normal components are zero.
Even (Flux Normal)	**Magnetic Field** is normal to the boundary; its tangential components are zero.

5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the boundary to the selected object.

Assigning a Balloon Boundary Condition for a 2D Magnetostatic Solver

To set a balloon boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Balloon**.
   The **Balloon Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

Assigning an Independent Boundary for a 2D Magnetostatic Solver

Independent and dependent boundaries enable you to model planes of periodicity where the **Magnetic Field** at every point on the dependent boundary surface is forced to match the **Magnetic Field** of every corresponding point on the independent boundary surface.

To set a independent boundary:

1. Select the edge to which you want to assign the independent boundary.
2. Click **Maxwell2D>Boundaries>Assign>Matching>Independent**.
   The **Independent Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. To reverse the direction of the vector, select the **Reverse Direction** check box.
5. Click **OK**.

Assigning a Dependent Boundary for a 2D Magnetostatic Solver

Assigning a dependent boundary is the second step in creating matching boundaries. The field on the independent boundary is mapped to the dependent boundary.

Note	You must define a independent boundary before creating the dependent boundaries that are associated with it. Assign dependent boundaries only to the outer surfaces of the problem region.

To set a dependent boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Matching>Dependent**.
   The **Dependent Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.

4. Select the independent boundary to which the dependent is to be assigned from the **Independent** pull-down list.
5. In the **Relation** section, select one of the following to specify the field behavior on the boundary:

Bdep = Bind	Click this radio button if the magnetic fields on the dependent and independent boundaries have the same magnitude and direction.
Bdep = -Bind	Click this radio button if the magnetic fields on the dependent boundary have the same magnitude as but the opposite direction from the independent boundary.

6. To reverse the direction of the vector, select the **Reverse Direction** check box.
7. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the boundary to the selected object.

2D Magnetostatic Excitations

The following excitations of magnetic fields are available for magnetostatic problems:

Excitation	Type of Excitation
Current Density	The known current density distribution in the cross-section of a conductor.
Current	The total current in a conductor.
Winding With Current	Current for both a stranded and solid conductor.
Winding With Resistance Voltage Drop	Resistance Voltage Drop for both a stranded and solid conductor.

In addition, permanent magnets materials serve as sources of magnetic fields.

Assigning a Current Density Excitation for a 2D Magnetostatic Solver

This command (depending upon solution type, either **X-Y** or **R-Z**) specifies the applicable components of the current density in the cross-section of an object. If the current density is a function of position, the value is entered in ampere/m^2, even if you change the units in the problem.

To define the current density:

1. Select the section of the geometry (i.e., the conductor) on which you want to apply the excitation (typically a 2D object).
2. Click **Maxwell2D>Excitations>Assign>Current Density**.
   The **Current Density Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, specify the following options:
   a. Enter a value for the current density in the **Value** box. The value may be a single numeric quantity or a functional quantity.
   b. If a functional quantity is entered in the **Value** box, which depends on the applicable coordinates, select a **Coordinate System** from the pull-down list.
5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the excitation to the selected object.

Assigning a Current Excitation for a 2D Magnetostatic Solver

Specifies the total current in the cross-section of an object.

To set a current excitation:

1. Select the section of the geometry (i.e., the conductor) on which you want to apply the excitation (typically a face or other 2D planar object).
2. Click **Maxwell2D>Excitations>Assign>Current**.
   The **Current Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, specify the following options:
   a. Enter a value for the current in the **Value** box, and select the units from the pull-down list.
   b. Select **Positive** or **Negative** as the **Ref. Direction**.
5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the excitation to the selected object.

Assigning a Winding Setup for a 2D Magnetostatic Solver

Note	• If you wish to define a winding for a 2D Eddy Current solution type, refer to "Assigning a Winding Setup for a 2D Eddy Current Solver " on page 12-41 . • If you wish to define a winding for a 2D Transient solution type, refer to Assigning a Winding Setup for a 2D Transient Solver.

You can use the predefined coil terminal(s) to define one or more current or voltage windings.

To define a winding for your model:

1. Click **Maxwell 2D>Excitations>Add Winding** to open the **Winding** dialog box.
2. Enter a name for the winding in the **Name** box, or accept the default.

3. In the **Parameters** section, select **Current**, or **Resistance Voltage Drop** from the **Type** drop-down menu.
4. Select the **Solid** or **Stranded** radio button to specify the type of conductor.
5. Enter values in the following fields, and select the desired units:

For a current winding	For a resistance voltage drop winding[1]
Current	Resistance
	Voltage

Note	You can also type a function as an expression for any of these items.

6. Enter a value in the **Number of parallel branches** text box.

Note	Parallel branches for solid windings assume there are no circulating currents in the parallel branches.

7. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the excitation to the selected object

The definition of a winding is final only after specifying the coils which belong to it.

Note

To add a coil terminal to this winding, do the following:

1. In the project tree, right-click the winding, and select **Add Coils**.
   The **Add Terminals** dialog box appears, listing all coil terminals that do not already belong to that winding.
2. Select the coil terminal (s) you want to add. To select multiple terminals, press **CTRL** and click each terminal.
3. Click **OK**.

To assign a coil terminal excitation and add it to this winding, do the following:

1. Select the section of the geometry on which you want to apply the coil terminal excitation.
2. In the project tree, right-click the winding, and select **Assign Coil**.
   The **Coil Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the number of conductors for the coil in the **Number of Conductors** box.

   Note: This value represents only the number of conductors inside of the selected geometry. If the Coil Terminal is cut due to symmetry, then only enter the Number of Conductors in the portion modeled.
5. Select one of the following for the **Polarity**:

   • **Positive**
   • **Negative**
   • **Function**

	6. If you selected **Function** as the **Polarity**, enter a function in the text box. 7. Optionally, click **Use Defaults** to revert to the default values in the dialog box. 8. Click **OK** to assign the coil terminal excitation. The coil terminal excitation is assigned and is added to the winding. To delete all coil terminal excitations that belong to this winding, do the following: • In the project tree, right-click the winding, and select **Delete All Coils**. All coil terminal excitations are removed from the winding and deleted from the excitations.

1. For a solid winding, the resistance term can represent: the resistance of a portion of the winding which is not modeled (for instance, end-effects), the leads connecting the winding to the source, or the source resistance. (The main winding resistance is calculated directly by the solver.) For a stranded winding, the resistance term is the complete DC resistance of the winding (since the solver does not determine resistance of a stranded winding) as well as the resistance of the end-effects, leads, source, etc. For both solid and stranded windings, the inductance term can represent: the extra inductance for a portion of the winding of the winding which is not modeled (for instance, end-effects), the leads connecting the winding to the source, or the source inductance. (The main winding inductance itself is calculated directly by the solver.)

Assigning a Coil for a 2D Magnetostatic Solver

To assign a coil as a magnetostatic excitation:

1. Select the section of the geometry on which you want to apply the excitation (typically a 2D object).
2. Click **Maxwell2D>Excitations>Assign>Coil**. (You can also right-click an existing winding in the project tree, and select **Assign Coil**.)
   The **Coil Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the number of conductors for the coil in the **Number of Conductors** box.
5. Select one of the following for the **Polarity**:
   - **Positive**
   - **Negative**
   - **Function**
6. If you selected **Function** as the **Polarity**, enter a function in the text box.
7. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the coil to the selected object.

Note	To add this coil to an existing winding, do the following:

	1. In the project tree, right-click the coil terminal, and select **Add to Winding**. The **Add to Winding** dialog box appears. 2. Select the winding to which you ant to add the coil terminal, and click **OK**.

Note	To remove a coil terminal from an existing winding, do the following: • In the project tree, right-click the coil terminal, and select **Remove from Winding**. The terminal is removed from the winding and moved up one level in the project tree (directly beneath **Excitations**). • Alternatively, all coils assigned to a winding may be deleted by right-clicking the winding in the project tree and selecting **Delete All Coils**.

Related Topics

"Assigning a Winding Setup for a 2D Magnetostatic Solver " on page 12-13

2D Electrostatic Boundaries and Excitations

Specify at least one of the following excitations as a source of electric fields:

- The charge on a surface or object.
- The charge density inside an object.
- Define the electric potential on each surface and object using a voltage excitation.

Include at least one of the following as a reference for computing the electric potential:

- A voltage excitation.
- An odd symmetry (flux normal) boundary.

Related Topics

2D Electrostatic Boundaries

2D Electrostatic Excitations

Technical Notes: Electric Field Calculation

Specifying the Solver Type

2D Electrostatic Boundaries

The following boundary conditions are available for electrostatic problems:

Boundary Type	E-Field Behavior	Used to model...
Default Boundary	Field behaves as follows: • Natural boundaries — The	Ordinary E-field behavior on boundaries. Object interfaces

Conditions (Natural and Neumann)	normal component of **D** changes by the amount of surface charge density, tangent component of **E** is continuous. No special conditions are imposed. • Neumann boundaries — **E** is tangential to the boundary. Flux cannot cross a Neumann boundary.	are initially set to natural boundaries; outer boundaries are initially set to Neumann boundaries.
Symmetry	Field behaves as follows: • Even Symmetry (Flux Tangential) — **E** is tangential to the boundary; its normal components are zero. • Odd Symmetry (Flux Normal) — **E** is normal to the boundary; its tangential components are zero.	Planes of geometric and electrical symmetry.
Balloon	Two options are available: Charge — The charge at "infinity" balances the charge in the drawing region. The net charge is zero. (Use for capacitance calculations) Voltage — The voltage at "infinity" is zero.	Electrically insulated structures (Charge option) or electrically grounded structures (Voltage option). For the voltage case, the balloon boundary will not be equipotential in general.
Matching (Independent and Dependent)	The E-field vector on the dependent boundary is forced to match the magnitude and direction (or the negative of the direction) of the E-field vector on the independent boundary.	Planes of symmetry in periodic structures where **E** is oblique to the boundary.

Default Boundary Conditions for a 2D Electrostatic Solver

These boundary conditions are automatically defined for an electrostatic model:

- Natural boundaries are assigned to the interfaces between objects. The normal component of **D** changes by the amount of surface charge density, tangent component of **E** is continuous.
- Neumann boundaries are assigned to the outside edges of the problem region. **E** is tangential to the boundary. Flux cannot cross a Neumann boundary.

To leave a surface set to its default boundary condition, do nothing. Deleted boundary conditions and excitations automatically revert to the default boundary conditions.

Assigning a Symmetry Boundary Condition for a 2D Electrostatic Solver

This boundary condition defines a plane of geometric or electric symmetry in a structure. Assign it only to the outer surfaces of the problem region.

To set a symmetry boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Symmetry**.
   The **Symmetry Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select one of the following as the type of symmetry:

Odd (Flux Normal)	**E** is normal to the boundary; its tangential components are zero.
Even (Flux Tangential)	**E** is tangential to the boundary; its normal components are zero.

5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the boundary to the selected object.

Assigning a Balloon Boundary Condition for a 2D Electrostatic Solver

To set a balloon boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Balloon**.
   The **Balloon Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select **Voltage** or **Charge** as the **Balloon Type**.
5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the boundary to the selected object.

Assigning a Independent Boundary for a 2D Electrostatic Solver

Independent and dependent boundaries enable you to model planes of periodicity where the E-field vector at every point on the dependent boundary surface is forced to match the E-field vector of every corresponding point on the independent boundary surface. The transformation used to map the E-field from the independent to the dependent is determined by specifying the orientation of a local, predefined coordinate system, indicated by an arrow, on both the independent and dependent boundaries.

To set an independent boundary:

1. Select the edge to which you want to assign the independent boundary.
2. Click **Maxwell2D>Boundaries>Assign>Matching>Independent**.
   The **Independent Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. To reverse the direction of the vector, select the **Reverse Direction** check box.
5. Click **OK**.

Assigning a Dependent Boundary for a 2D Electrostatic Solver

Assigning a dependent boundary is the second step in creating matching boundaries. The field on the Independent boundary is mapped to the dependent boundary.

Note	You must define an Independent boundary before creating the dependent boundaries that are associated with it. Assign dependent boundaries only to the outer surfaces of the problem region.

To set a dependent boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Matching>Dependent**.
   The **Dependent Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select the independent boundary to which the dependent is to be assigned from the **Independent** pull-down list.
5. In the **Relation** section, select one of the following to specify the field behavior on the boundary:

Edep = Eind	Click this radio button if the dependent and independent boundaries have the same magnitude and direction.
Edep = -Eind	Click this radio button if the dependent boundary field has the same magnitude as but the opposite direction from the

	independent boundary field.

6. To reverse the direction of the vector, select the **Reverse Direction** check box.
7. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the boundary to the selected object.

2D Electrostatic Excitations

The following excitations are available for electrostatic problems:

Excitation	Type of Excitation
Voltage	The DC voltage on a surface or object.
Charge	The total charge on a surface or object.
Floating	Used to model conductors at unknown potentials.
Charge Density	The charge density over an object (dielectric).

Assigning a Voltage Excitation for a 2D Electrostatic Solver

Voltage excitations are identical to voltage boundaries in previous versions of Maxwell. A voltage excitation can be assigned to any surfaces or 2D objects. These assignments could include spatial functions.

This type of excitation sets the electric potential to a specific value.

To set a voltage excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell2D>Excitations>Assign>Voltage**.
   The **Voltage Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the electric potential in the **Value** box, and select the units from the pull-down list. You can enter a numerical value, parameter (design or project variable) or a spatial function.
5. For a spatial function, select a coordinate system from the **Coordinate System** pull-down list.
6. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
7. Click **OK** to assign the excitation to the selected object.

Assigning a Charge Excitation for a 2D Electrostatic Solver

This type of excitation defines the total charge on a surface or object. The potential distribution on the object with charge excitation is computed during the solution.

To define a charge excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell2D>Excitations>Assign>Charge**.
   The **Charge Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the numerical value, parameter (design or project variable) or a spatial function in the **Value** box.
5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the excitation to the selected object.

Assigning a Floating Excitation for a 2D Electrostatic Solver

This type of excitation models conductors at unknown potentials and specifies the total charge on the conductor.

To define a floating conductor:

1. Select the section of the geometry on which you want to apply the excitation. This can be a 2D or 1D geometry entity. If it is a 2D object, it doesn't have to be a conductor (but it will behave like one when the solution becomes available).
2. Click **Maxwell2D>Excitations>Assign>Floating**.
   The **Floating Excitation** dialog box appears.
3. Click the **General** tab.
4. Enter a name for the excitation in the **Name** box, or accept the default.
5. In the **Parameters** section, enter the charge value or parameter in the **Value** box.
6. Optionally, click **Use Defaults** to revert to the default values in the dialog box. You can also click the **Defaults** tab to save new default settings or to revert to the standard default values.
7. Click **OK** to assign the excitation to the selected object.

Assigning a Charge Density Excitation for a 2D Electrostatic Solver

This type of excitation defines the volume charge density on an object.

To define the charge density on a dielectric object:

1. Select the object on which you want to apply the excitation.
2. Click **Maxwell2D>Excitations>Assign>Charge Density**.
   The **Charge Density Excitation** dialog box appears.

3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the charge density in the **Value** box. You can enter a numerical value, parameter (design or project variable) or a spatial function.
5. For a spatial function, select the type of coordinate system from the **Coordinate System** pull-down list.
6. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
7. Click **OK** to assign the excitation to the selected object.

2D AC Conduction Boundaries and Excitations

The following excitations can be specified as the source of electric fields:

- The voltage difference between surfaces and objects. Define the electric potential on each surface and object using a voltage excitation.

Include at least one of the following as a reference for computing the electric potential:

- A voltage excitation.
- An odd symmetry (flux normal) boundary.

Related Topics

2D AC Conduction Boundaries

2D AC Conduction Excitations

Technical Notes: AC Conduction Field Calculation

Specifying the Solver Type

2D AC Conduction Boundaries

The following boundary conditions are available for AC conduction electric problems:

Boundary Type	E-Field Behavior	Used to model...
Default Boundary Conditions (Natural and Neumann)	Field behaves as follows: • Natural boundaries - The normal component of **J** and the tangential component of **E** are continuous across discontinuity surfaces. No special conditions are imposed. • Neumann boundaries - **E** and **J** vectors are tangential to the boundary. Flux cannot cross a Neumann boundary.	Object interfaces are initially set to natural boundaries; outer boundaries are initially set to Neumann boundaries.
Symmetry	Field behaves as follows: • Even Symmetry (Flux Tangential) - **E and J**	Planes of geometric and electrical symmetry.

	vectors are tangential to the boundary; its normal components are zero. • Odd Symmetry (Flux Normal) - **E and J vectors are** normal to the boundary; its tangential components are zero.	
Balloon	Balloon boundaries model the region outside the drawing space as being nearly “infinitely” large — effectively isolating the model from other voltage sources.	Electrically insulated structures, equipotential lines can cross this type of boundary. In general, they are neither tangent nor perpendicular to the balloon boundary.
Matching (Independent and Dependent)	The E-field on the dependent boundary is forced to match the magnitude of the E-field on the independent boundary if the "plus" sign is used. If the "minus" sign is used, the field on the dependent boundary will oscillate with opposite phase angle (180 degrees phase shift).	Planes of symmetry in periodic structures where **E** is oblique to the boundary.

Default Boundary Conditions for a 2D AC Conduction Solver

These boundary conditions are automatically defined for a AC conduction model:

- Natural boundaries are assigned to the surfaces between objects. The normal component of **J** and the tangential component of **E** are continuous across discontinuity surfaces.
- Neumann boundaries are assigned to the outside edges of the problem region. **E** and **J** vectors are tangential to the boundary. Flux cannot cross a Neumann boundary.

To leave a surface set to its default boundary condition, do nothing. Deleted boundary conditions and excitations automatically revert to the default boundary conditions.

Assigning a Symmetry Boundary Condition for a 2D AC Conduction Solver

This boundary condition defines a plane of geometric or electric symmetry in a structure. Assign it only to the outer surfaces of the problem region.

To set a symmetry boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Symmetry**.
   The **Symmetry Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.

4. Select one of the following as the type of symmetry:

Odd (Flux Normal)	Voltages have opposite sign on the two sides of the symmetry plane.
Even (Flux Tangential)	Voltages have same sign on the two sides of the symmetry plane.

5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the boundary to the selected object.

Assigning a Balloon Boundary Condition for a 2D AC Conduction Solver

To set a balloon boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Balloon**.
   The **Balloon Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

Assigning an Independent Boundary for a 2D AC Conduction Solver

Independent and dependent boundaries enable you to model planes of periodicity where the **E**-field at every point on the dependent boundary surface is forced to match the **E**-field of every corresponding point on the independent boundary surface.

To set an independent boundary:

1. Select the edge to which you want to assign the independent boundary.
2. Click **Maxwell2D>Boundaries>Assign>Matching>Independent**.
   The **Independent Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. To reverse the direction of the vector, select the **Reverse Direction** check box.
5. Click **OK**.

Maxwell computes the E-field on this boundary and maps it to the dependent boundary.

Assigning a Dependent Boundary for a 2D AC Conduction Solver

Assigning a dependent boundary is the second step in creating matching boundaries. The field on the Independent boundary is mapped to the dependent boundary.

Note	You must define an Independent boundary before creating the dependent boundaries

	that are associated with it. Assign dependent boundaries only to the outer surfaces of the problem region.

To set a dependent boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Matching>Dependent**.
   The **Dependent Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. In the **Parameters** section, select the independent boundary to which the dependent is to be assigned from the **Independent** pull-down list.
5. Under **Relation**, select one of the following to specify the field behavior on the boundary:

Edep = Eind	Click this radio button if the dependent and independent boundaries have the same magnitude and oscillate in phase.
Edep = -Eind	Click this radio button if the dependent boundary field has the same magnitude as the independent boundary field and oscillate 180 degrees out of phase.

6. To reverse the direction of the vector, select the **Reverse Direction** check box.
7. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the boundary to the selected object.

2D AC Conduction Excitations

The following excitations are available for electric AC conduction problems:

Excitation	Type of Excitation
Voltage	The AC voltage on a surface or object. This excitation type is available for all three of the 2D electric solvers.

Note	All field quantities oscillate with the same frequency, equal to the frequency specified in the solution setup. The phase of quantities computed in the solution space however will be different in general.

Assigning a Voltage Excitation for a 2D AC Conduction Solver

This type of excitation sets the electric potential (voltage) on a surface or object to a specific value.

To set a voltage excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell2D>Excitations>Assign>Voltage**.
   The **Voltage Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the electric potential in the **Value** box, and select the units from the pull-down list. You can enter a numerical value or a spatial function.
5. For a spatial function, select a coordinate system from the **Coordinate System** pull-down list.
6. Enter a value in the **Phase** text box, and select a unit from the pull-down list.
7. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the excitation to the selected object.

2D DC Conduction Boundaries and Excitations

The following excitations can be specified as the source of electric fields:

- The voltage difference between surfaces and objects. Define the electric potential on each surface and object using a voltage excitation.

Include at least one of the following as a reference for computing the electric potential:

- A voltage excitation.
- An odd symmetry (flux normal) boundary.

Related Topics

2D DC Conduction Boundaries

2D DC Conduction Excitations

Technical Notes: DC Conduction Field Calculation

Specifying the Solver Type

2D DC Conduction Boundaries

The following boundary conditions are available for DC conduction electric problems:

Boundary Type	E-Field Behavior	Used to model...
Default Boundary Conditions (Natural and Neumann)	Field behaves as follows: • Natural boundaries — Normal component of **J** and tangential component of **E** are continuous across discontinuity surfaces. No special conditions are imposed. • Neumann boundaries — **E** and **J** vectors are tangential to the boundary. Flux	Object interfaces are initially set to natural boundaries; outer boundaries are initially set to Neumann boundaries.

	cannot cross a Neumann boundary.	
Symmetry	Field behaves as follows: • Even Symmetry (Flux Tangential) — **E** and **J** vectors are tangential to the boundary; their normal components are zero. • Odd Symmetry (Flux Normal) — **E** and **J** vectors are normal to the boundary; their tangential components are zero.	Planes of geometric and electrical symmetry.
Balloon	Balloon boundaries model the region outside the drawing space as being nearly “infinitely” large — effectively isolating the model from other voltage sources.	Electrically insulated structures.
Resistance	A resistance boundary models a very thin layer of resistive material (such as that caused by deposits, coatings or oxidation on a metallic surface) on a conductor at a known potential.	Use this boundary condition when the resistive layer’s thickness is much smaller than the other dimensions of the model.
Matching (Independent and Dependent)	The **E**-field on the dependent boundary is forced to match the magnitude and direction (or the negative of the direction) of the **E**-field on the independent boundary.	Planes of symmetry in periodic structures where **E** is oblique to the boundary.

Default Boundary Conditions for a 2D DC Conduction Solver

These boundary conditions are automatically defined for a DC conduction model:

- Natural boundaries are assigned to the surfaces between objects. Normal component of **J** and tangential component of **E** are continuous across discontinuity surfaces. No special conditions are imposed.
- Neumann boundaries are assigned to the outside edges of the problem region. **E** and **J** vectors are tangential to the boundary. Flux cannot cross a Neumann boundary.

To leave a surface set to its default boundary condition, do nothing. Deleted boundary conditions and excitations automatically revert to the default boundary conditions.

Assigning a Symmetry Boundary Condition for a 2D DC Conduction Solver

This boundary condition defines a plane of geometric or electric symmetry in a structure. Assign it only to the outer surfaces of the problem region.

To set a symmetry boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Symmetry**.
   The **Symmetry Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select one of the following as the type of symmetry:

Odd (Flux Normal)	Voltages have opposite signs on the two sides of the symmetry plane.
Even (Flux Tangential)	Voltages have same signs on the two sides of the symmetry plane.

5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the boundary to the selected object.

Assigning a Balloon Boundary Condition for a 2D DC Conduction Solver

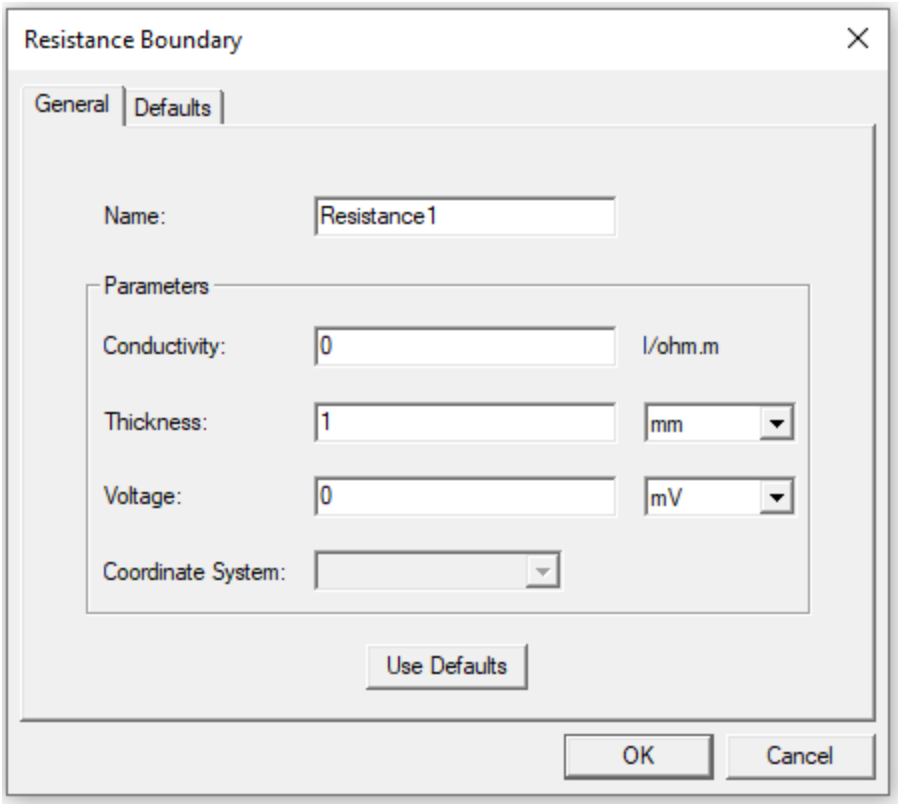
To set a balloon boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Balloon**.
   The **Balloon Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

Assigning a Resistance Boundary Condition for a 2D DC Conduction Solver

To set a resistance boundary:

1. Select the edge of the geometry on which you want to apply the boundary condition.
2. Click **Maxwell2D>Boundaries>Assign>Resistance**.
   The **Resistance Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Enter the boundary parameters in the **Conductivity**, **Thickness**, and **Voltage** fields, and select the units of measure. Use of variables is supported.



5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the boundary to the selected object.

Note	Apply resistance boundaries only to the outside edge of the problem space. The external region touching to a resistance boundary is assumed to be a PEC (perfect conductor) with a known potential value that is specified in the resistance boundary setup. Refer to *Resistance* for technical details.

Assigning an Independent Boundary for a 2D DC Conduction Solver

Independent and dependent boundaries enable you to model planes of periodicity where the **E**-field at every point on the dependent boundary surface is forced to match the **E**-field of every corresponding point on the independent boundary surface.

To set an independent boundary:

1. Select the edge to which you want to assign the independent boundary.
2. Click **Maxwell2D>Boundaries>Assign>Matching>Independent**.
   The **Independent Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. To reverse the direction of the vector, select the **Reverse Direction** check box.
5. Click **OK**.

Assigning a Dependent Boundary for a 2D DC Conduction Solver

Assigning a dependent boundary is the second step in creating matching boundaries. The field on the independent boundary is mapped to the dependent boundary.

Note	You must define an Independent boundary before creating the dependent boundaries that are associated with it. Assign dependent boundaries only to the outer surfaces of the problem region.

To set a dependent boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Matching>Dependent**.
   The **Dependent Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select the independent boundary to which the dependent is to be assigned from the **Independent** pull-down list.
5. In the **Relation** section, select one of the following to specify the field behavior on the boundary:

Edep = Eind	Click this radio button if the **E** field on the dependent and independent boundaries have the same magnitude and direction.
Edep = -Eind	Click this radio button if the **E** field on the dependent boundary field has the same magnitude as but the opposite direction from the independent boundary field.

6. To reverse the direction of the vector, select the **Reverse Direction** check box.
7. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the boundary to the selected object.

2D DC Conduction Excitations

The following excitations are available for electric DC conduction problems:

Excitation	Type of Excitation
Voltage	The DC voltage on a surface or object.

Assigning a Voltage Excitation for a 2D DC Conduction Solver

This type of excitation sets the electric potential (voltage) on a surface or object to a specific value.

To set a voltage excitation:

1. Select the section of the geometry on which you want to apply the excitation.
2. Click **Maxwell2D>Excitations>Assign>Voltage**.
   The **Voltage Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the electric potential in the **Value** box, and select the units from the pull-down list. You can enter a numerical value or a spatial function.
5. For a spatial function, select a coordinate system from the **Coordinate System** pull-down list.
6. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
7. Click **OK** to assign the excitation to the selected object.

2D Eddy Current Boundaries and Excitations

Specify at least one of the following as a source of AC magnetic fields in your model:

- The current in the cross-section (object), which can be either stranded or solid.
- The current to be divided among several parallel conductors.
- The current density in an object.
- The external magnetic field.

If currents or current densities are the only sources of AC magnetic fields in your model, set at least one outer boundary to the following:

- The default boundary conditions.
- An odd symmetry (flux tangential) boundary.
- An even symmetry (flux normal) boundary.

Related Topics

2D Eddy Current Boundaries

2D Eddy Current Excitations

Technical Notes: Eddy Current Field Simulation

Specifying the Solver Type

2D Eddy Current Boundaries

The eddy current field solver allows you to define the following types of boundaries:

Boundary Type	H-Field Behavior	Used to model…
Default Boundary Conditions (Natural and Neumann)	Field behaves as follows: • Natural boundaries — Tangential **H** and normal **B** are continuous across surfaces without current density distribution; tangential **H** has a jump if the surface has current density distribution. • Neumann boundaries — **Magnetic Field** is tangential to the boundary and flux cannot cross it.	Initially, object interfaces are natural boundaries; outer boundaries and excluded objects are Neumann boundaries.
Vector Potential	Sets the magnetic vector potential $A_Z(t)$, or $rA_\phi(t)$, on the boundary. The behavior of H depends on whether $A_Z(t)$ or $rA_\phi(t)$ is constant or functional.	Magnetically isolated structures.
Impedance	Includes the effect of induced currents beyond the boundary surface based on tangent component of **H** along the impedance boundary.	Conductors with very small skin depths.
Symmetry	Field behaves as follows: • Odd Symmetry (Flux Tangential) — **Magnetic Field**is tangential to the boundary; its normal components are zero. • Even Symmetry (Flux Normal) — **Magnetic Field**is normal to the boundary; its tangential components are zero.	Planes of geometric and magnetic symmetry.
Balloon	Models the case where the structure is “infinitely” far away from other magnetic fields or current sources.	Magnetically isolated structures.
Matching (Independent and Dependent)	The **Magnetic Field** on the dependent boundary is forced to match the magnitude and direction of the **Magnetic Field** on the independent boundary. If a 'plus" sign is used, the fields oscillate in phase, if a minus sign is used the fields will oscillate 180 degrees out of phase.	Planes of symmetry in periodic structures where **Magnetic Field**is oblique to the boundary.

Default Boundary Conditions for the 2D Eddy Current Solver

These boundary conditions are automatically defined for an eddy current model:

- Natural boundaries are assigned to the surfaces between objects. Tangential **H** and normal **B** are continuous across surfaces without current density distribution; tangential **H** has a jump if the surface has current density distribution.
- Neumann boundaries are assigned to the outside edges of the problem region. **Magnetic Field** is tangential to the boundary and flux cannot cross it.

To leave a surface set to its default boundary condition, do nothing. Deleted boundary conditions and excitations automatically reset to the default boundary conditions.

Assigning a Vector Potential for the 2D Eddy Current Solver

To define a vector potential boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Vector Potential**.
   The **Vector Potential Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the value of the potential. The value may be specified as a single numeric value or as a spatial function.
5. If a functional value is specified, select a **Coordinate System** from the pull-down list:
6. Enter a value in the **Phase** text box, and select a unit from the pull-down list.
7. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the boundary to the selected object.

The new boundary is added to the boundary list in the project tree.

Assigning an Impedance Boundary for the 2D Eddy Current Solver

This boundary condition is used to simulate the effect of induced currents in a conductor without explicitly computing them. Since the conductor must be excluded in the model (saving time needed to mesh and solve for currents), assign the impedance boundary condition to an outside edge of the problem region or to an excluded object. Exclude the object from the problem region and make the object a perfect conductor in the Material Manager.

To define an impedance boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Impedance**.
   The **Impedance Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Enter the conductivity (in inverse ohm-meters) in the **Conductivity** field.
5. Enter the conductor's relative permeability in the **Permeability** field.
6. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
7. Click **OK** to assign the boundary to the selected object.

Related Topics

Impedance Boundary

2D Impedance Boundary

Impedance boundaries allow you to simulate the effect of induced currents in a conductor without explicitly computing them. Use this boundary condition for models where:

- The skin depth in the conductor is less than two orders of magnitude smaller than the dimensions of the structure. In models like this, the meshmaker may not be able to create a fine enough mesh in the conductor to compute eddy currents.
- The magnetic field decays much more rapidly inside the conductor in the direction that's normal to the surface than it does in directions that are tangential to the surface.
- The AC current source is relatively far away from the surface where eddy currents occur, compared to the size of the skin depth.

The conductor itself must be excluded from the solution region. When setting up the model, do one of the following:

- For external boundaries, when drawing the model, make the surface along which eddy currents are to be computed an outer surface of the problem region.
- For internal boundaries: assign a perfect conductor to the object in the Material Manager, and also exclude the object from the solution by unchecking **Solve Inside** under the object attributes. The solver does not find solutions inside a perfect conductor.

Then, when defining boundaries, assign an impedance boundary to the individual surfaces of the problem region (for external boundaries) or to the entire object (for internal boundaries). By entering the conductivity, s, and the relative permeability, μ_r, of the object, you specify the skin depth of induced eddy currents. The simulator uses this skin depth value when computing the electromagnetic field solution. It assumes that the **Magnetic Field** falls off exponentially inside the conductor. The ohmic loss due to induced currents can then be computed from the **Magnetic Vector Potential** along the impedance boundary - on the surface of the object that you are interested in - according to the formula below or by using the quantity **EdgeLossDensity** in the 2D calculator.

$$EdgeLossDensity(W/m^2) = \pi f \sqrt{\frac{\pi f \sigma}{\mu_0 \mu_r}} [A_z \bullet A_z^*]$$

where:

A_z *= z component of the Magnetic Vector Potential*

A_z^* *= complex conjugate of the z component of the Magnetic Vector Potential*

μ_0 *= permeability of free space*

μ_r *= relative permeability of the impedance boundary*

σ *= conductivity of the impedance boundary (S/m)*

f = frequency (Hz)

You may also use **EdgeLossDensity** to plot the loss along an edge containing an impedance boundary.

Note	An impedance boundary only approximates the effect of eddy currents acting at a shallow skin depth. It does not directly compute them. In general, the fields modeled using an impedance boundary will closely match the field patterns that would actually occur in the structure. However, at discontinuities in the surface (such as corners), the field patterns may be different.

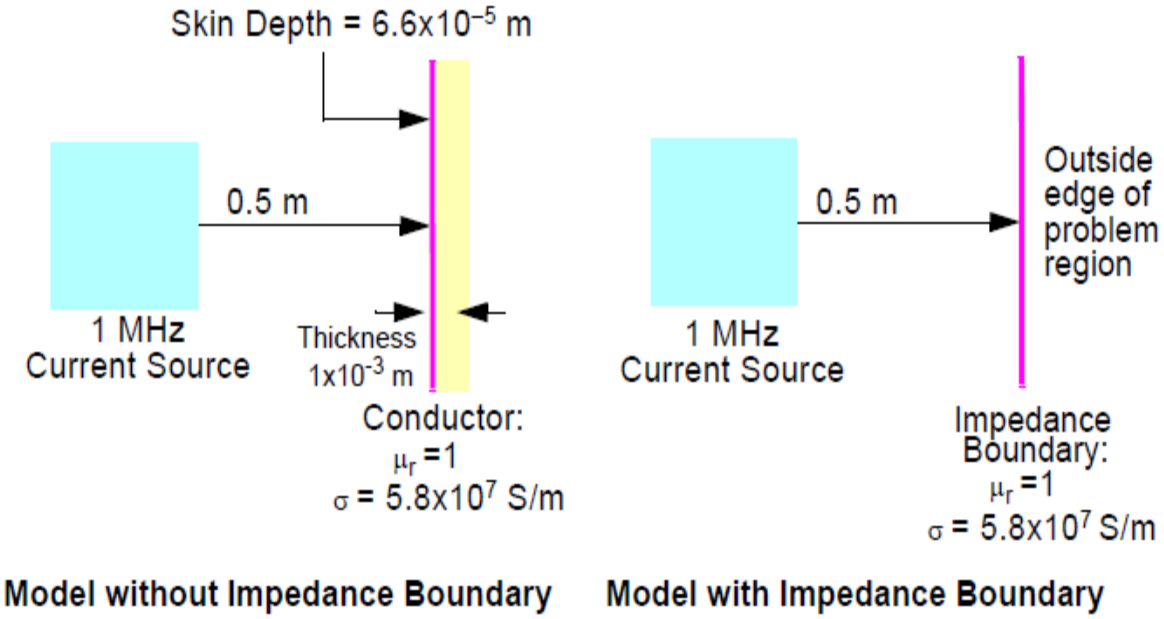
Related Topics

When to Use Impedance Boundaries

Assigning an Impedance Boundary for the 2D Eddy Current Solver

When to Use Impedance Boundaries in 2D

A typical situation where impedance boundaries can be used to reduce the complexity of a model is shown below. Suppose you want to compute eddy current losses in the conductor next to the current source shown below on the left. If the source carries AC current at a frequency of 1 MHz, the skin depth in the conductor is 6.6 x 10–5 meters. This is several orders of magnitude smaller than the conductor's thickness. Since the conductor where currents are induced is also relatively far away from the current source, an impedance boundary can be used to model the induced currents — as shown on the right



The conductor must be excluded from the model by making it a perfect conductor. The outside boundary of the model is moved to the inside surface of the conductor. This outside surface is defined as an impedance boundary, using the conductivity and permeability specified previously. Since the simulator does not have to actually compute a solution inside the conductor, the field

solution is computed more quickly and uses less memory. After solving, you can compute the ohmic loss for the surface using the solution calculator and plot the loss density on the boundary.

Related Topics

Assigning an Impedance Boundary for the 2D Eddy Current Solver

Assigning a Symmetry Boundary for the 2D Eddy Current Solver

This boundary condition defines a plane of geometric or magnetic symmetry in a structure. Assign it only to the outer surfaces of the problem region.

To set a symmetry boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Symmetry**.
   The **Symmetry Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select one of the following as the type of symmetry:

Odd (Flux Tangential)	Magnetic field is tangential to the boundary; its normal components are zero.
Even (Flux Normal)	Magnetic field is normal to the boundary; its tangential components are zero.

5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the boundary to the selected object.

Related Topics

Setting a Symmetry Multiplier

Assigning a Balloon Boundary Condition for the 2D Eddy Current Solver

To set a balloon boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Balloon**.
   The **Balloon Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

Assigning an Independent Boundary for the 2D Eddy Current Solver

Independent and dependent boundaries enable you to model planes of periodicity where the **Magnetic Field** at every point on the dependent boundary surface is forced to match the **Magnetic Field** of every corresponding point on the independent boundary surface. The transformation used to map the **Magnetic Field** from the independent to the dependent is determined by specifying a coordinate system on both the independent and dependent boundaries.

To set an independent boundary:

1. Select the edge to which you want to assign the independent boundary.
2. Click **Maxwell2D>Boundaries>Assign>Matching>Independent**.
   The **Independent Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. To reverse the direction of the vector, select the **Reverse Direction** check box.
5. Click **OK**.

Assigning a Dependent Boundary for the 2D Eddy Current Solver

Assigning a dependent boundary is the second step in creating matching boundaries. The field on the independent boundary is mapped to the dependent boundary.

Note	You must define an independent boundary before creating the dependent boundaries that are associated with it. Assign dependent boundaries only to the outer surfaces of the problem region.

To set a dependent boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Matching>Dependent**.
   The **Dependent Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select the independent boundary to which the dependent is to be assigned from the **Independent** pull-down list.
5. In the **Relation** section, select one of the following to specify the field behavior on the boundary:

Bdep = Bind	Click this radio button if the magnetic fields on the dependent and independent boundaries have the same magnitude and the fields oscillate in phase.
Bdep = -Bind	Click this radio button if the magnetic fields on the dependent boundary have the same

	magnitude as the independent boundary field but the fields oscillate 180 degrees out of phase.

6. To reverse the direction of the vector, select the **Reverse Direction** check box.
7. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the boundary to the selected object.

2D Eddy Current Excitations

The eddy current solver allows you to define the following sources of AC magnetic fields:

Excitation	Type of Excitation
Current	The total current in a conductor.
Parallel Current	The total current to be divided among several conductors in parallel. The method of dividing the current depends upon the choice of solid or stranded.
Current Density	The current density in a conductor. In this case, you must also define the current density terminals for the conductor.
Coil	Used to define one or more model windings.
End Connection	End connection is used to model details of the "squirrel cage" winding of induction machines, dampers in synchronous machines, etc.
Winding With Current	Current for both a stranded and solid conductor.
Winding With Voltage	Voltage for both a stranded and solid conductor.
Winding With External Circuit Connection	External circuit connection for both a stranded and solid conductor.

Assigning a Current Excitation for the 2D Eddy Current Solver

Specifies the total AC current in a conduction path. The conduction path may be contained completely within the problem region (for example, a coil), or may touch the edges of the problem region.

To specify the total AC current:

1. Select the 2D section of the geometry on which you want to apply the excitation.
2. Click **Maxwell2D>Excitations>Assign>Current**.
   The **Current Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. Enter the current strength in the **Value** box.
5. Enter the phase for the current in the **Phase** box.
6. Select the **Solid** or **Stranded** radio button to specify the type of conductor.
7. Select either **Positive** or **Negative** as the **Ref. Direction** for the conductor.
8. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
9. Click **OK** to assign the excitation to the selected object.

Assigning a Parallel Current Excitation for the 2D Eddy Current Solver

Specifies the total AC current in a parallel conduction path consisting of at least two conductors.

To specify the total AC current:

1. Select the 2D sections of the geometry on which you want to apply the excitation. At least two section must be specified
2. Click **Maxwell2D>Excitations>Assign>Parallel Current**.
   The **ParallelCurrent Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. Enter the current strength in the **Value** box.
5. Enter the phase for the current in the **Phase** box.
6. Select the **Solid** or **Stranded** radio button to specify the type of conductor.

Note	For parallel solid conductors, the current split will be based upon the overall field solution including eddy effects (the currents in individual parallel conductors will have complex values, which are proportional with the respective admittances, and in general will have different phase angles between them). If stranded conductors are specified, the current is split based upon the relative areas of the selected conductors.

7. Select either **Positive** or **Negative** as the **Ref. Direction** for the conductor.
8. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
9. Click **OK** to assign the excitation to the selected object.

Assigning a Current Density Excitation for the 2D Eddy Current Solver

This command specifies the applicable component (depending on XY or RZ solution type) of the AC current density. If the current density is a function of position, the value is entered in ampere/m^2, even if you change the units in the problem.

To define the current density:

1. Select the section of the geometry (i.e., the conductor) in which you want to apply the excitation (typically a 2D object).
2. Click **Maxwell2D>Excitations>Assign>Current Density**.
   The **Current Density Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, specify the following options:
   a. Enter the current strength in the **Value** box. The value may be a single numeric value or a spatial function.
   b. If the value is specified as a spatial function, select the **Coordinate System** from the pull-down list.
   c. Enter the phase angle, θ, of the current density in the **Phase** field, and select a unit from the pull-down list.
5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the excitation to the selected object.

Assigning a Coil for a 2D Eddy Current Solver

To assign a coil as an Eddy current excitation:

1. Select the section of the geometry on which you want to apply the excitation (typically a 2D object).
2. Click **Maxwell2D>Excitations>Assign>Coil**. (You can also right-click an existing winding in the project tree, and select **Assign Coil**.)

   The **Coil Excitation** dialog box appears.
3. On the **General** tab, enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the number of conductors for the coil in the **Number of Conductors** box.
5. Select one of the following for the **Polarity**:
   - **Positive**
   - **Negative**
   - **Function**
6. If you selected **Function** as the **Polarity**, enter a function in the text box.
7. Optionally, on the Defaults tab, click **Save Defaults** to save the current settings or click **Revert to Standard Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the coil to the selected object.

Note	To add this coil to an existing winding, do the following: 1. In the project tree, right-click the coil terminal, and select **Add to Winding**. The **Add to Winding** dialog box appears. 2. Select the winding to which you ant to add the coil terminal, and click **OK**.

Note	To remove a coil terminal from an existing winding, do the following:

	• In the project tree, right-click the coil terminal, and select **Remove from Winding**. The terminal is removed from the winding and moved up one level in the project tree (directly beneath **Excitations**). • Alternatively, all coils assigned to a winding may be deleted by right-clicking the winding in the project tree and selecting **Delete All Coils**.

Assigning an End Connection for a 2D Eddy Current Solver

End connection is used to model details of the "squirrel cage" winding of induction machines, dampers in synchronous machines, etc.

To define an end connection:

1. Select the section of the geometry (i.e., the conductor) across which you want to apply the excitation.
2. Click **Maxwell 2D>Excitations>Assign>End Connection**.

   The **End Connection Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, specify the following options:
   a. Enter the **End resistance between adjacent conductors**, and select a unit of measurement from the pull-down list.
   b. Enter the **End inductance between adjacent conductors**, and select a unit of measurement from the pull-down list.
5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the excitation to the selected object.

Assigning a Winding Setup for a 2D Eddy Current Solver

Note	• If you wish to define a winding for a 2D transient solution type, refer to "Assigning a Winding Setup for a 2D Transient Solver " on page 12-54 . • If you wish to define a winding for a 2D magnetostatic solution type, refer to Assigning a Winding Setup for a 2D Magnetostatic Solver.

You can use the pre-defined coil terminal(s) to define one or more current or voltage windings.

To define a winding for your model:

1. Click **Maxwell 2D>Excitations>Add Winding** to open the **Winding** dialog box.
2. Enter a name for the winding in the **Name** box, or accept the default.
3. In the **Parameters** section:
   a. Select **Current**, **Voltage**, or **External** from the **Type** pull-down list. The winding type selected enables only those fields (Initial Current, Resistance, Inductance, Voltage,

Phase) applicable for the chosen type.

b. Select the **Solid** or **Stranded** radio button to specify the type of conductor.

c. Enter values in the following fields (enabled according to winding type), and select the desired units:

For a current winding	For a voltage winding[1]	For an external winding
Current	Resistance	(no settings required)
Phase	Inductance	
	Voltage	
	Phase	

Note	You can also type a function as an expression for any of these fields except Initial Current.

d. Enter a value in the **Number of parallel branches** text box.

Note	Parallel branches for solid windings assume there are no circulating currents in the parallel branches.

4. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
5. Optionally, on the Defaults tab, click **Save Defaults** to save the current settings or click **Revert to Standard Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the excitation to the selected object.

The definition of a winding is final only after specifying the coils which belong to it.

Note	To add a coil terminal to this winding, do the following: 1. In the project tree, right-click the winding, and select **Add Coils**. The **Add Terminals** dialog box appears, listing all coil terminals that do not already belong to that winding. 2. Select the coil terminal (s) you want to add. To select multiple terminals, press **CTRL** and click each terminal. 3. Click **OK**. To assign a coil terminal excitation and add it to this winding, do the following: 1. Select the section of the geometry on which you want to apply the coil terminal excitation. 2. In the project tree, right-click the winding, and select **Assign Coil**. The **Coil Excitation** dialog box appears. 3. Enter a name for the excitation in the **Name** box, or accept the default. 4. In the **Parameters** section, enter the number of conductors for the coil in the **Number of Conductors** box. **Note:** This value represents only the number of conductors inside of the selected geometry. If the Coil Terminal is cut due to symmetry, then only enter the Number of Conductors in the portion modeled.

	5. Select one of the following for the **Polarity**: • **Positive** • **Negative** • **Function** 6. If you selected **Function** as the **Polarity**, enter a function in the text box. 7. Optionally, click **Use Defaults** to revert to the default values in the dialog box. 8. Click **OK** to assign the coil terminal excitation. The coil terminal excitation is assigned and is added to the winding. To delete all coil terminal excitations that belong to this winding, do the following: • In the project tree, right-click the winding, and select **Delete All Coils**. All coil terminal excitations are removed from the winding and deleted from the excitations.

Related Topics

"Assigning a Coil for a 2D Eddy Current Solver " on page 12-40

1. For a solid winding, the resistance term can represent: the resistance of a portion of the winding which is not modeled (for instance, end-effects), the leads connecting the winding to the source, or the source resistance. (The main winding resistance is calculated directly by the solver.) For a stranded winding, the resistance term is the complete DC resistance of the winding (since the solver does not determine resistance of a stranded winding) as well as the resistance of the end-effects, leads, source, etc. For both solid and stranded windings, the inductance term can represent: the extra inductance for a portion of the winding of the winding which is not modeled (for instance, end-effects), the leads connecting the winding to the source, or the source inductance. (The main winding inductance itself is calculated directly by the solver.)

External Circuit Setup for 2D Eddy Current

The Maxwell Circuit design editor is used to set up the external circuit excitation for windings in Maxwell 2D and 3D Eddy current solutions. Because the Eddy current solver is a frequency domain solver, only linear circuit components are supported, which include Resistance, Inductance, Capacitance, and Transformer. In addition, only sinusoidal sources should be used as sources. Sinusoidal Current Source and Sinusoidal Voltage Source, designed for transient solutions, can also be used for eddy current solutions. For example, to use the sinusoidal current source, you should set the parameters as follows:

- Set **I0**, **Td**, and **Df** to zero, where I0 is offset current in Amps, Td is the delay time in seconds, and Df is the damping factor in 1/seconds.
- Set **TIME** as the **Type**.
- Use **Ia** to set the peak amplitude in Amps.

- Use **IFreq** to set the signal frequency. Note that setting IFreq is not necessary as it will be internally set to the frequency defined in the Maxwell design setup.
- Use **Phase** to set the phase delay.

With these specifications, the time signal can be written as:

$$I(t) = I_a \sin\left(2\pi I_{freq} - Phase\right)$$

Note that “Phase” is the signal phase delay. The complex source in the eddy current solver is represented in phasor form as:

$$I = I_a \; e^{j\theta}$$

and the time signal can be written in terms of (I_a, θ) as:

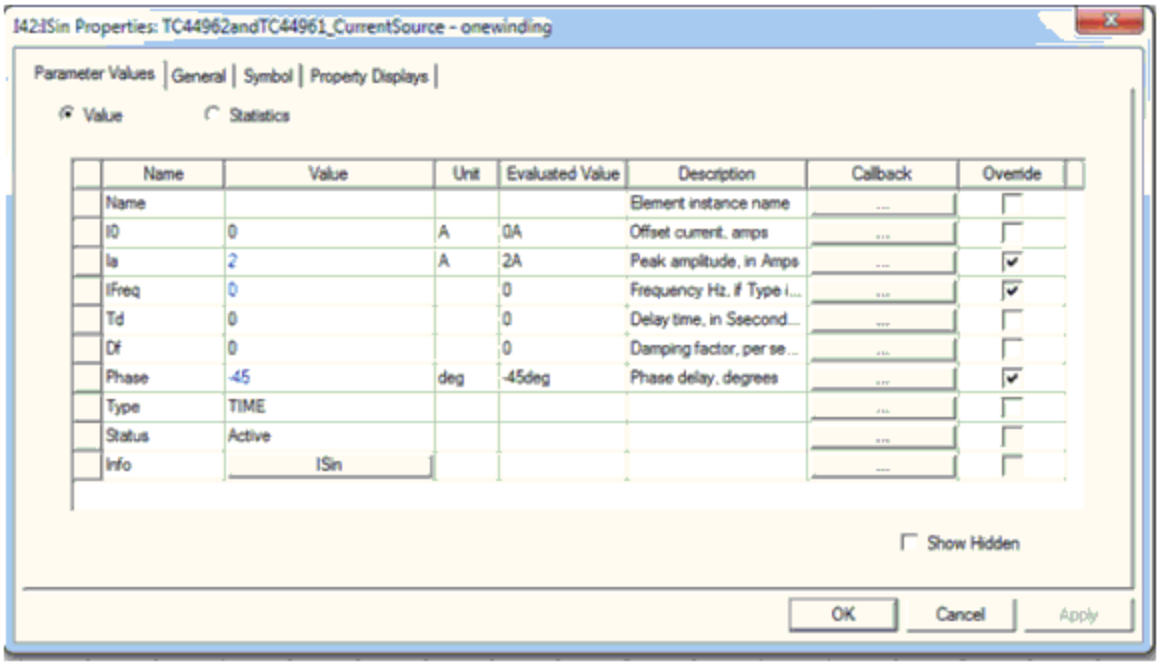
$$I(t) = Im\left\{I_a \; e^{j(2\pi I_{freq}+\theta)}\right\}$$

In the complex expression, θ is defined as phase, while in Maxwell Circuit, *Phase* is defined as phase delay. Thus:

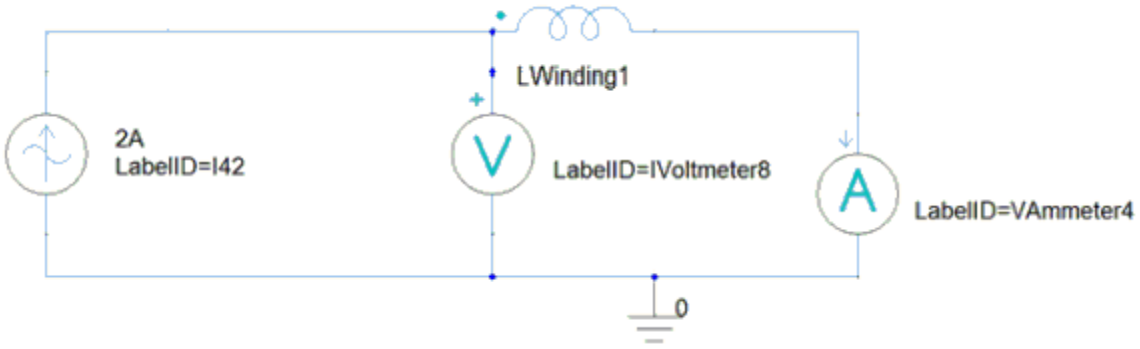
$$Phase = -\theta$$

As a result, you should enter the negative of phase of the complex signal as the Phase value.

An example of such a setup for $I = 2e^{j45^\circ}$ is shown below.



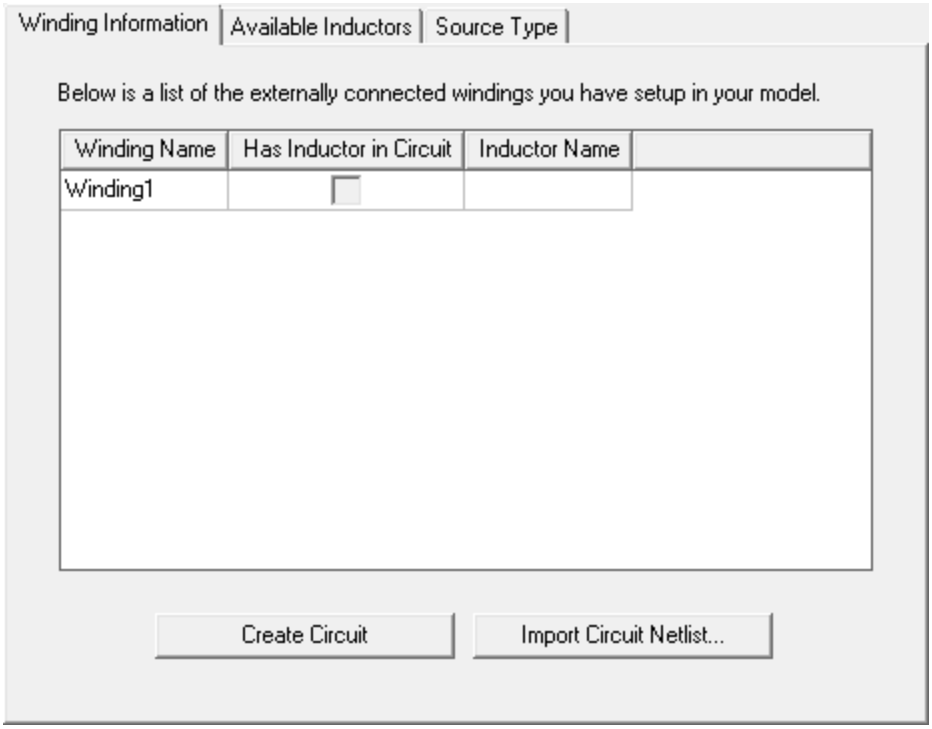
An example of an external circuit excitation for the winding for an Eddy current solver is shown below.



Note	For frequency domain eddy current solutions, only linear circuit components are supported, which include Resistor, Inductor, Capacitor, and Transformer. Only sinusoidal sources should be used as sources.

To set up an external circuit connection for an External winding type:

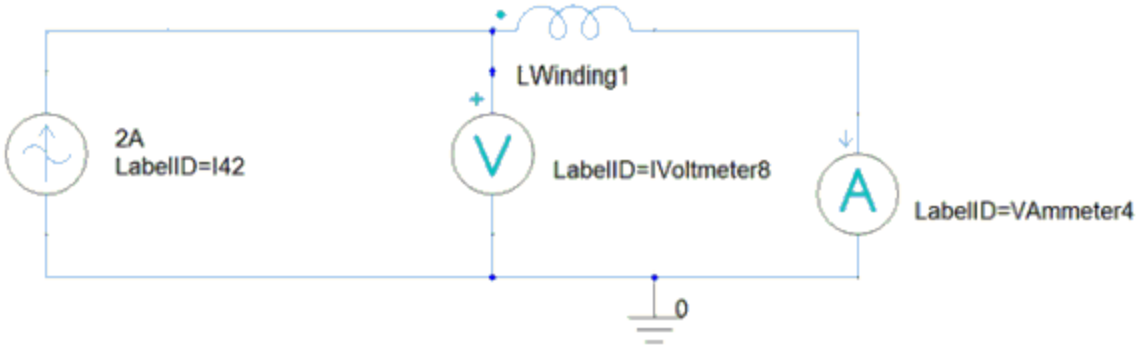
1. Click **Maxwell 2D>Excitations>External Circuit>Edit External Circuit** to open the **Edit External Circuit** dialog box. You can also right-click on **Excitations** in the Project tree and select **External Circuit>Edit External Circuit**.
    a. If no external winding connections have been set up for your model, the **Winding Information** tab lists the external winding(s) in the design. Continue with step 2.



b. If external windings have already been set up, continue with step 7.

2. On the **Winding Information** tab, click **Create Circuit** to add a circuit design to the project. The new circuit design automatically opens for editing in the Maxwell Circuit Editor. The circuit includes one inductor for each external winding in the original Maxwell transient design. The following example is for a design having one external winding.

3. Add the components needed to complete the external circuit design such as shown the following example.



4. When the circuit design is finished, click **Maxwell Circuit>Export Netlist** and export the netlist **.sph** file to the desired location.
5. Return to the Maxwell design and click **Maxwell 2D>Excitations>External Circuit>Edit External Circuit** to open the **Edit External Circuit** dialog box. You can also right-click on **Excitations** in the Project tree and select **External Circuit>Edit External Circuit**.
6. On the **Winding Information** tab, click **Import Circuit** to import the circuit.
7. To view a list of inductors in the imported circuit, click the **Available Inductors** tab.
8. To view the sources used in the externally connected windings, click the **Source Type** tab.
9. If variables are used in the imported circuit, they and their design values are listed on the **Parameter Values** tab. You can link to variables within the imported circuit as follows:
    a. On the **Parameter Values** tab, click in the **Value** column of the parameter to be mapped.

b. Enter a new local variable name and press the `Enter` key to open the **Add Variable** dialog.

c. Select a **Unit Type** and **Unit**, and enter a **Value** for the variable and click **OK**. The new Maxwell variable is now mapped to the variable in the imported circuit, and can be varied directly by Maxwell or used for Optimetrics analyses.

10. If the imported file is a .**sph** file, you can click the **Circuit Path** tab to view the original project and design names.

Note	After you import a circuit netlist, the **Create Circuit** button changes to **Update Netlist from Circuit**. Use **Update Netlist from Circuit** to reimport the netlist to keep it current.

11. Click **OK** to close the **Edit External Circuit** dialog box.

Related Topics

"Assigning a Winding Setup for a 2D Eddy Current Solver " on page 12-41

2D Transient Boundaries and Excitations

Define at least one of the following as a source of magnetic fields:

- The stranded or solid windings, with a voltage or current supply or connected to an external circuit.
- Permanent magnet(s).

You will need to set at least one outer boundary to the following:

- Value (vector potential) boundary conditions.
- An odd symmetry (flux tangential) boundary.
- An even symmetry (flux normal) boundary.

Related Topics

2D Transient Boundaries

2D Transient Excitations

Technical Notes: Transient Simulation

Specifying the Solver Type

Permanent Magnet Excitations

2D Transient Boundaries

The transient field solver allows you to define the same types of boundary conditions as magnetostatic ones.

Boundary Type	H-Field Behavior	Used to model...

Default Boundary Conditions (Natural and Neumann)	Field behaves as follows: • Natural boundaries — Tangential **H** and normal **B** are continuous across surfaces without current density distribution. Tangential **H** has a jump if the surface has current density distribution. • Neumann boundaries — Magnetic field is normal to the boundary.	Initially, object interfaces have natural boundaries; outer boundaries, and excluded objects have Neumann boundaries.
Vector Potential	Sets the magnetic vector potential A_z , or rA_ϕ , on the boundary. The behavior of H depends on whether A_z or rA_ϕ is constant or functional.	Outer boundaries at specific vector potentials; externally applied magnetic fields.
Symmetry	Field behaves as follows: • Odd Symmetry (Flux Tangential) — **Magnetic Field** is tangential to the boundary; its normal components are zero. • Even Symmetry (Flux Normal) — **Magnetic Field** is normal to the boundary; its tangential components are zero.	Planes of geometric and magnetic symmetry.
Balloon	Models the case where the structure is “infinitely” far away from other magnetic fields, permanent magnets, or current sources.	Magnetically isolated structures.
Matching (Independent and Dependent)	The **Magnetic Field** on the dependent boundary is forced to match the magnitude and direction (or the negative of the direction) of the **Magnetic Field** on the independent boundary.	Planes of symmetry in periodic structures where **Magnetic Field**is oblique to the boundary.

Default Boundary Conditions for a 2D Transient Solver

These boundary conditions are automatically defined for a transient model:

- Natural boundaries are assigned to the surfaces between objects. Tangential **H** and normal **B** are continuous across surfaces without current density distribution. Tangential **H** has a jump if the surface has current density distribution.
- Neumann boundaries are assigned to the outside edges of the problem region. Magnetic field is normal to the boundary.

To leave a surface set to its default boundary condition, you do not need to set any boundary conditions. Deleted boundary conditions and excitations automatically reset to the default boundary conditions.

Assigning a Vector Potential for a 2D Transient Solver

To define a vector potential boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge). **Maxwell 2D** requires that the Vector Potential be assigned on a 1D object such as an edge or a line.
2. Click **Maxwell2D>Boundaries>Assign>Vector Potential**.
   The **Vector Potential Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the value of the potential. The value may be specified as a single numeric value or a spatial function.
5. If a functional value is specified, select a **Coordinate System** from the pull-down list:
6. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
7. Click **OK** to assign the boundary to the selected object.

The new boundary is added to the boundary list in the project tree.

Assigning a Symmetry Boundary for a 2D Transient Solver

This boundary condition defines a plane of geometric or magnetic symmetry in a structure. Assign it only to the outer surfaces of the problem region.

To set a symmetry boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Symmetry**.
   The **Symmetry Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Select one of the following as the type of symmetry:

Odd (Flux Tangential)	**Magnetic Field** is tangential to the boundary; its normal components are zero.
Even (Flux Normal)	**Magnetic Field** is normal to the boundary; its tangential components are zero.

5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the boundary to the selected object.

Related Topics

Setting a Symmetry Multiplier

Assigning a Balloon Boundary Condition for a 2D Transient Solver

To set a balloon boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Balloon**.
   The **Balloon Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. Click **OK** to assign the boundary to the selected object.

Assigning an Independent Boundary for a 2D Transient Solver

Independent and dependent boundaries enable you to model planes of periodicity where the **Magnetic Field** at every point on the dependent boundary surface is forced to match the **Magnetic Field** of every corresponding point on the independent boundary surface.

To set an independent boundary:

1. Select the edge to which you want to assign the independent boundary.
2. Click **Maxwell2D>Boundaries>Assign>Matching>Independent**.
   The **Independent Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.
4. To reverse the direction of the vector, select the **Reverse Direction** check box.
5. Click **OK**.

Maxwell computes the **Magnetic Field** on this boundary and maps it to the dependent boundary using the transformation defined by the independent and dependent coordinate systems.

Assigning a Dependent Boundary for a 2D Transient Solver

Assigning a dependent boundary is the second step in creating matching boundaries. The field on the independent boundary is mapped to the dependent boundary.

Note	You must define an independent boundary before creating the dependent boundaries that are associated with it. Assign dependent boundaries only to the outer surfaces of the problem region.

To set a dependent boundary:

1. Select the section of the geometry on which you want to apply the boundary condition (typically an edge).
2. Click **Maxwell2D>Boundaries>Assign>Matching>Dependent**.
   The **Dependent Boundary** dialog box appears.
3. Enter a name for the boundary in the **Name** box, or accept the default.

4. Select the independent boundary to which the dependent is to be assigned from the **Independent** pull-down list.
5. In the **Relation** section, select one of the following to specify the field behavior on the boundary:

Bdep = Bind	Click this radio button if the magnetic field on the dependent and independent boundaries have the same magnitude and direction.
Bdep = -Bind	Click this radio button if the magnetic field on the dependent boundary has the same magnitude as but the opposite direction from the independent boundary field.

6. To reverse the direction of the vector, select the **Reverse Direction** check box.
7. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the boundary to the selected object.

2D Transient Excitations

Transient problems use the following types of excitations:

Excitation	Type of Excitation
Current	The total current in a conductor
Current Density	The known current density distribution in a conductor.
Coil	Used to define one or more model windings.
End Connection	End connection is used to model details of the "squirrel cage" winding of induction machines, dampers in synchronous machines, etc.
Winding With Current	Current for both a stranded and solid conductor.
Winding With Voltage	Voltage for both a stranded and solid conductor.
Winding With External Circuit Connection	External circuit connection for both a stranded and solid conductor.

Note	A winding is a versatile type of excitation. Six different combinations can be used: Current, Voltage, and External Circuit — each with a solid or stranded conductor. The Coil Terminal type of excitation needs to be defined prior to setting up a winding.

In addition, permanent magnet materials serve as sources of magnetic fields.

Assigning a Current Excitation for a 2D Transient Solver

Specifies the total transient current in an object.

To specify the total transient current:

1. Select the 2D section of the geometry on which you want to apply the excitation.
2. Click **Maxwell2D>Excitations>Assign>Current**.
   The **Current Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. Enter the current magnitude, time dependent, in the **Value** box, and select a unit from the pull-down list.
5. Select the **Solid** or **Stranded** radio button to specify the type of conductor.
6. Select either **Positive** or **Negative** as the **Ref. Direction** for the conductor.
7. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the excitation to the selected object.

Assigning a Current Density Excitation for a 2D Transient Solver

This command specifies the applicable component (depending on the XY or RZ type of application) of the transient current density in an object. If the current density is a function of position, the value is entered in ampere/m^2, even if you change the units in the problem.

To define the current density:

1. Select the section of the geometry (i.e., the conductor) in which you want to apply the excitation (typically a 2D object).
2. Click **Maxwell2D>Excitations>Assign>Current Density**.
   The **Current Density Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, specify the following options:
   a. Enter the current strength in the **Value** box. The value may be a single numeric value or a spatial function.
   b. If a spatial function is entered for the **Value**, select the **Coordinate System** from the pull-down list.
5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the excitation to the selected object.

Assigning a Coil for a 2D Transient Solver

To assign a coil as a transient excitation:

1. Select the section of the geometry on which you want to apply the excitation (typically a 2D object).

2. Click **Maxwell2D>Excitations>Assign>Coil**. (You can also right-click an existing winding in the project tree, and select **Assign Coil**.)
   The **Coil Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the number of conductors for the coil in the **Number of Conductors** box.
5. Select one of the following for the **Polarity**:
   - **Positive**
   - **Negative**
   - **Function**
6. If you selected **Function** as the **Polarity**, enter a function in the text box.
7. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the coil to the selected object.

Note	To add this coil to an existing winding, do the following: 1. In the project tree, right-click the coil terminal, and select **Add to Winding**. The **Add to Winding** dialog box appears. 2. Select the winding to which you ant to add the coil terminal, and click **OK**.

Note	To remove a coil terminal from an existing winding, do the following: • In the project tree, right-click the coil terminal, and select **Remove from Winding**. The terminal is removed from the winding and moved up one level in the project tree (directly beneath **Excitations**). • Alternatively, all coils assigned to a winding may be deleted by right-clicking the winding in the project tree and selecting **Delete All Coils**.

Related Topics

"Assigning a Winding Setup for a 2D Eddy Current Solver " on page 12-41

Assigning an End Connection for a 2D Transient Solver

End connection is used to model details of the "squirrel cage" winding of induction machines, dampers in synchronous machines, etc.

To define an end connection:

1. Select the section of the geometry (i.e., the conductor) across which you want to apply the excitation.
2. Click **Maxwell 2D>Excitations>Assign>End Connection**.
   The **End Connection Excitation** dialog box appears.

3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, specify the following options:
   a. Enter the **End resistance between adjacent conductors**, and select a unit of measurement from the pull-down list.
   b. Enter the **End inductance between adjacent conductors**, and select a unit of measurement from the pull-down list.
5. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
6. Click **OK** to assign the excitation to the selected object.

Related Topics

Technical Notes: End Connections in 2D Transient

Assigning a Winding Setup for a 2D Transient Solver

Note	• If you wish to define a winding for a 2D Eddy Current solution type, refer to "Assigning a Winding Setup for a 2D Eddy Current Solver " on page 12-41 . • If you wish to define a winding for a 2D Magnetostatic solution type, refer to Assigning a Winding Setup for a 2D Magnetostatic Solver.

You can use the pre-defined coil terminal(s) to define one or more current or voltage windings.

To define a winding for your model:

1. Click **Maxwell 2D>Excitations>Add Winding** to open the **Winding** dialog box.
2. Enter a name for the winding in the **Name** box, or accept the default.
3. In the **Parameters** section, select **Current**, **Voltage**, or **External** from the **Type** drop-down menu.

Note	Maxwell transient designs can be dynamically coupled to Twin Builder components through the Twin Builder user interface. You can enable this feature on the Advanced Product Coupling tab of the **Design Settings** dialog box. Source windings set to **External** are available as conservative pins in Twin Builder. For more information, refer to the Help for Twin Builder.

4. Select the **Solid** or **Stranded** radio button to specify the type of conductor.
5. Enter values in the following fields, and select the desired units:

For a current winding	For a voltage winding[1]	For an external winding
Current	Initial Current	Initial Current
	Resistance	
	Inductance	
	Voltage	

Note	You can also type a function as an expression for any of these items except initial current.

6. Enter a value in the **Number of parallel branches** text box.

Note	Parallel branches for solid windings assume there are no circulating currents in the parallel branches.

7. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the excitation to the selected object

The definition of a winding is final only after specifying the coils which belong to it.

Note

To add a coil terminal to this winding, do the following:

1. In the project tree, right-click the winding, and select **Add Coils**.
   The **Add Terminals** dialog box appears, listing all coil terminals that do not already belong to that winding.
2. Select the coil terminal (s) you want to add. To select multiple terminals, press **CTRL** and click each terminal.
3. Click **OK**.

To assign a coil terminal excitation and add it to this winding, do the following:

1. Select the section of the geometry on which you want to apply the coil terminal excitation.
2. In the project tree, right-click the winding, and select **Assign Coil**.
   The **Coil Excitation** dialog box appears.
3. Enter a name for the excitation in the **Name** box, or accept the default.
4. In the **Parameters** section, enter the number of conductors for the coil in the **Number of Conductors** box.

   Note: This value represents only the number of conductors inside of the selected geometry. If the Coil Terminal is cut due to symmetry, then only enter the Number of Conductors in the portion modeled.
5. Select one of the following for the **Polarity**:
   - **Positive**
   - **Negative**
   - **Function**
6. If you selected **Function** as the **Polarity**, enter a function in the text box.
7. Optionally, click **Use Defaults** to revert to the default values in the dialog box.
8. Click **OK** to assign the coil terminal excitation.
   The coil terminal excitation is assigned and is added to the winding.

To delete all coil terminal excitations that belong to this winding, do the following:

- In the project tree, right-click the winding, and select **Delete All Coils**.
  All coil terminal excitations are removed from the winding and deleted from the excitations.

Related Topics

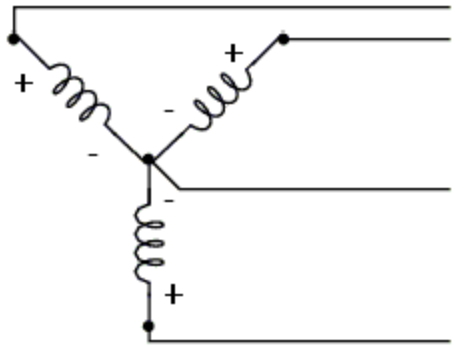
Setting Up a Y Connection in 2D and 3D Transient Designs

---

1. For a solid winding, the resistance term can represent: the resistance of a portion of the winding which is not modeled (for instance, end-effects), the leads connecting the winding to the source, or the source resistance. (The main winding resistance is calculated directly by the solver.) For a stranded winding, the resistance term is the complete DC resistance of the winding (since the solver does not determine resistance of a stranded winding) as well as the resistance of the end-effects, leads, source, etc. For both solid and stranded windings, the inductance term can represent: the extra inductance for a portion of the winding of the winding which is not modeled (for instance, end-effects), the leads connecting the winding to the source, or the source inductance. (The main winding inductance itself is calculated directly by the solver.)

Setting Up a Y Connection in 2D and 3D Transient Designs

The Y Connection function available in 2D and 3D Transient solution types allows multiple windings to be connected in a classical Y (sometimes referred to as wye) configuration with the negative terminals connected to a common node as illustrated below.



Setup Y Connection requires voltage winding definitions be set and the negative voltage terminal for each winding will be connected to form the common node of the circuit.

Each voltage source specified in the winding description drives the positive side of the coil. Y Connections are commonly used in motor and transformer applications driven by 3-phase voltage; however, any number of windings may be connected to a common node in Maxwell.

Note	• Only one Y Connection is allowed in a given Maxwell

	design. • A Y Connection requires use of stranded voltage windings. • Duplicate windings are not allowed.

To define a Y-connection:

1. Click **Maxwell 2D** (or **Maxwell 3D**)**>Excitations>Setup Y Connection**.
   The **Setup Y Connection** dialog appears.
2. In the **Winding** list on the left, select the windings to be grouped into a Y connection.
3. Click the **Group** button to create a Y Connection group with the selected windings.
4. Click **OK** when all required grouping have been created.

Related Topics

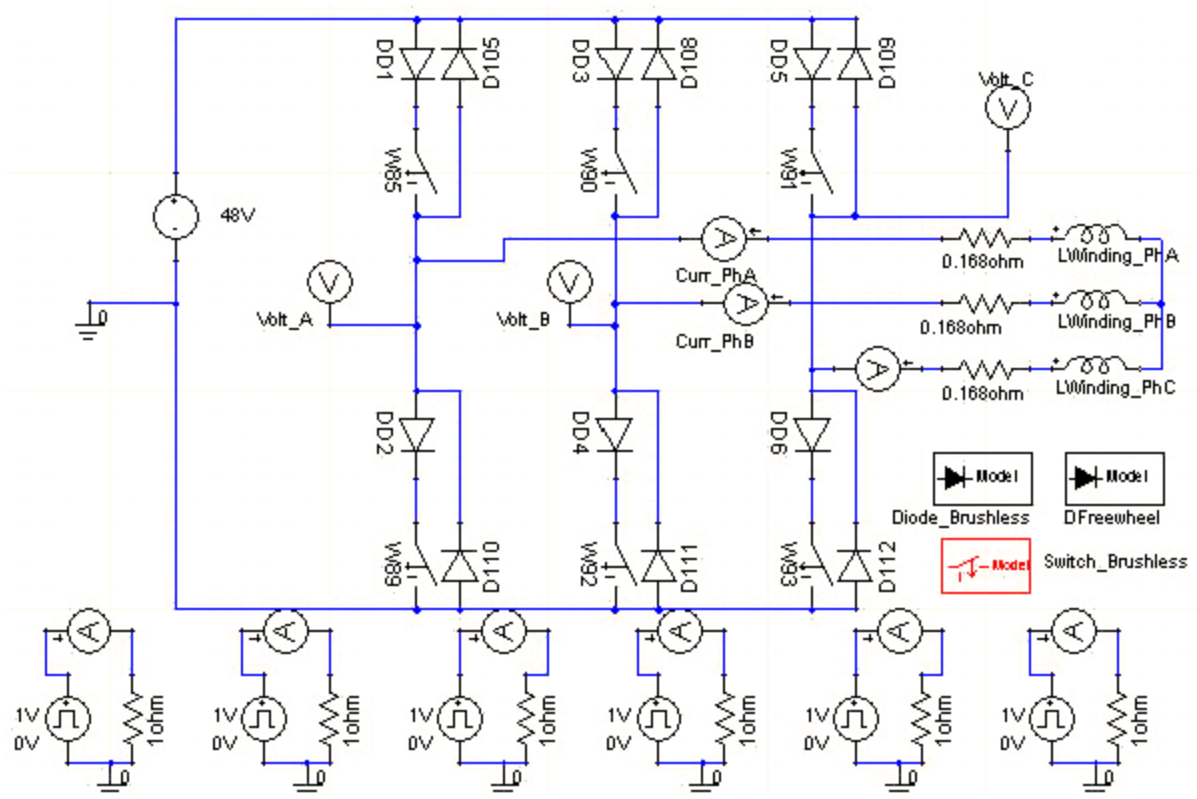
Assigning a Winding Setup for a 2D Transient Solver
Assigning a Winding Setup for a 3D Transient Solver

Setting Up an External Circuit in 2D Transient

Note	Use of external circuits is indicated when more complex excitation circuits such as rectifiers, inverters, etc., containing components such as diodes, transistors, complex sources, etc. are needed. There is no need to use an external circuit for simple current or voltage sources supplying the windings.

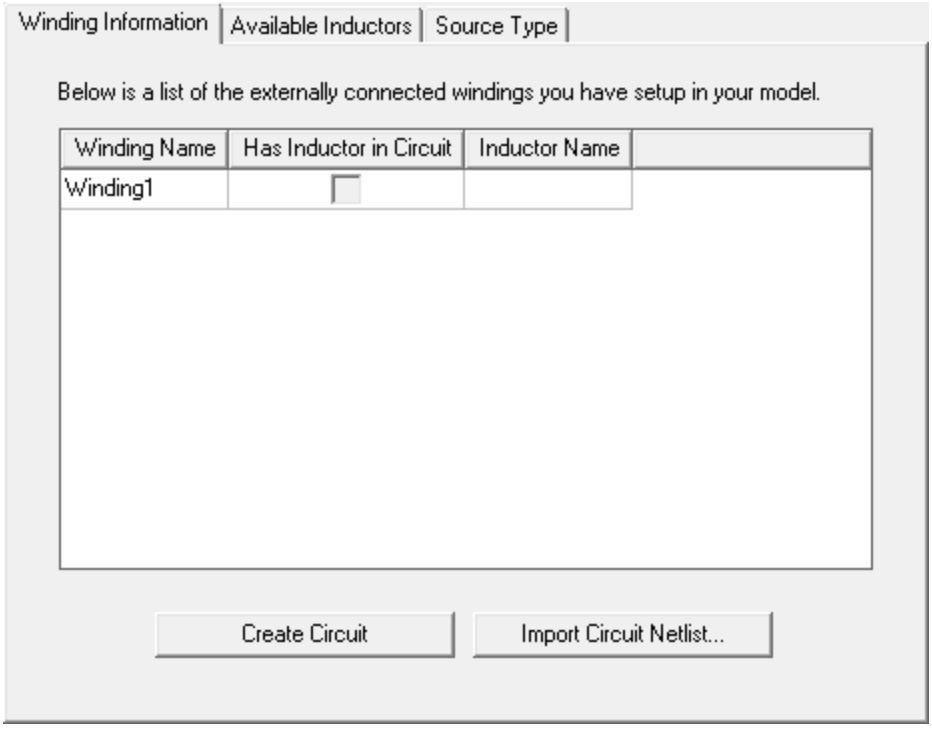
Note	External circuits intended for use with the Time Decomposition Method (TDM) can only support basic circuit elements that are solution-independent. This is because, for TDM, all time steps (or all time steps in a subdivision) are solved simultaneously - and thus the circuit topology and its parameters must be known before solving. Therefore some circuit elements, such as solution-dependent switches, diodes, capacitors, and inductors which are not directly connected to windings in series, are not supported.

An external circuit netlist can be complex as shown in the example below.



To set up an external circuit connection for an External winding type:

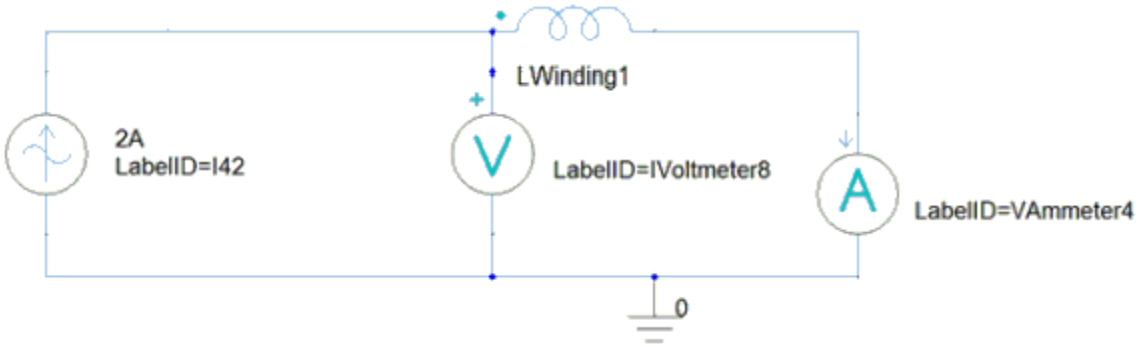
1. Click **Maxwell 2D>Excitations>External Circuit>Edit External Circuit** to open the **Edit External Circuit** dialog box. You can also right-click on **Excitations** in the Project tree and select **External Circuit>Edit External Circuit**.
    a. If no external winding connections have been set up for your model, the **Winding Information** tab lists the external winding(s) in the design. Continue with step 2.



b. If external windings have already been set up, continue with step 7.

2. On the **Winding Information** tab, click **Create Circuit** to add a circuit design to the project. The new circuit design automatically opens for editing in the Maxwell Circuit Editor. The circuit includes one inductor for each external winding in the original Maxwell transient design. The following example is for a design having one external winding.

3. Add the components needed to complete the external circuit design such as shown the following example.



4. When the circuit design is finished, click **Maxwell Circuit>Export Netlist** and export the netlist **.sph** file to the desired location.
5. Return to the Maxwell design and click **Maxwell 2D>Excitations>External Circuit>Edit External Circuit** to open the **Edit External Circuit** dialog box. You can also right-click on **Excitations** in the Project tree and select **External Circuit>Edit External Circuit**.
6. On the **Winding Information** tab, click **Import Circuit** to import the circuit.
7. To view a list of inductors in the imported circuit, click the **Available Inductors** tab.
8. To view the sources used in the externally connected windings, click the **Source Type** tab.
9. If variables are used in the imported circuit, they and their design values are listed on the **Parameter Values** tab. You can link to variables within the imported circuit as follows:
    a. On the **Parameter Values** tab, click in the **Value** column of the parameter to be mapped.
    b. Enter a new local variable name and press the `Enter` key to open the **Add Variable** dialog.
    c. Select a **Unit Type** and **Unit**, and enter a **Value** for the variable and click **OK**. The new Maxwell variable is now mapped to the variable in the imported circuit, and can be varied directly by Maxwell or used for Optimetrics analyses.
10. If the imported file is a .**sph** file, you can click the **Circuit Path** tab to view the original project and design names.

Note	After you import a circuit netlist, the **Create Circuit** button changes to **Update Netlist from Circuit**. Use **Update Netlist from Circuit** to reimport the netlist to keep it current.

11. Click **OK** to close the **Edit External Circuit** dialog box.

Note	The user-set time steps can be modified when external circuits are used to drive the windings of the transient finite element model. Following is a list of situations that will lead to a time step change: • All power electronic switching instances that do not coincide with user-specified solve times request a new solution time from the transient solver. The respective switching generating a new solve time request for the transient solver can be time,

	position, or speed dependent. • When current and/or voltage sources with a piecewise linear variation are used, a new solution time is requested from the transient solver at each (time) definition point used in the corresponding source definition table. • When the change in any winding inductance value is excessive, a new time step (smaller) is calculated and a new solution generated (re-calculated) accordingly. If a user-specified save field time is missed because of the above reasons, the next solved time step fields are saved instead.

Related Topics

"Assigning a Winding Setup for a 2D Transient Solver " on page 12-54

"Setting Up an External Circuit Using Simulink " on page 11-109

"Setting Up an External Circuit Using the Maxwell Circuit Editor " on page 27-79

Setting Up an External Circuit for 2D Transient using Simulink

In addition to using the Maxwell Circuit Editor to set up an external circuit, you also can use Matlab® Simulink® to set up an external circuit for both Maxwell 2D and 3D transient designs.

- Supported Simulink versions are: R2017b, R2018a, R2018b, R2019a, R2019b, R2020a, R2020b, and R2021a.

Refer to "Setting Up an External Circuit Using Simulink " on page 11-109 for detailed information.

Related Topics

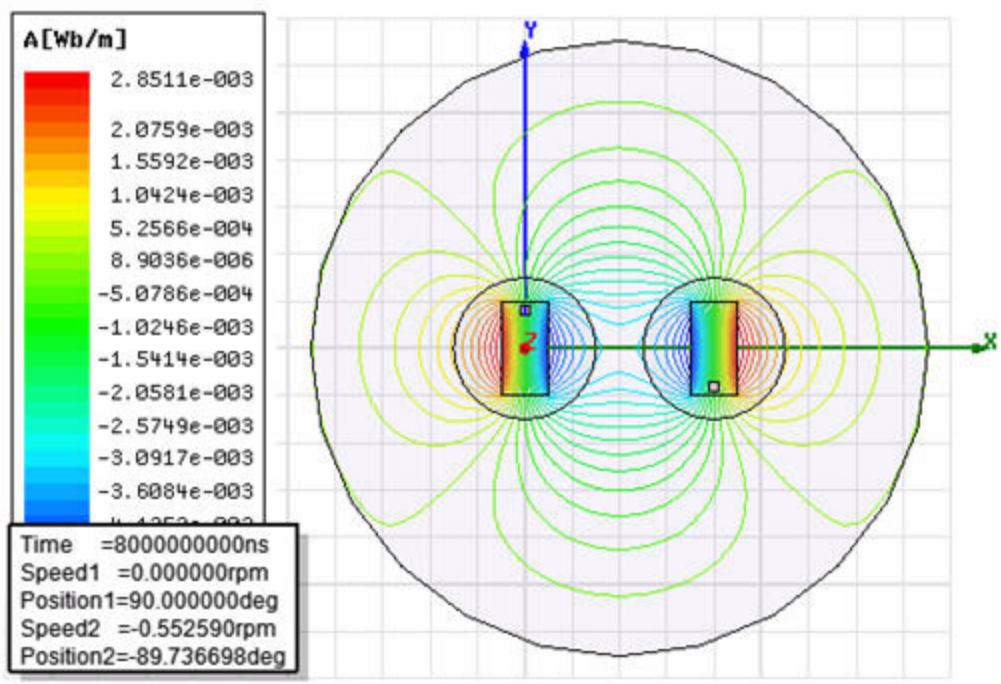
"Setting Up an External Circuit in 2D Transient " on page 12-57

13 - Setting Up Motion for Transient Projects

For projects where you selected transient as the solution type, you can define how the model components move in relation to one another. This is accomplished by assigning a **MotionSetup** to a **Band** object. Two or more motion bands may be assigned, each with its own independent motion characteristics. Multiple motion bands are used primarily for certain types of motors where multiple rotors may rotate about a common central axis; however, other applications are possible as shown in the following examples.

Example - Two Bands of Motion

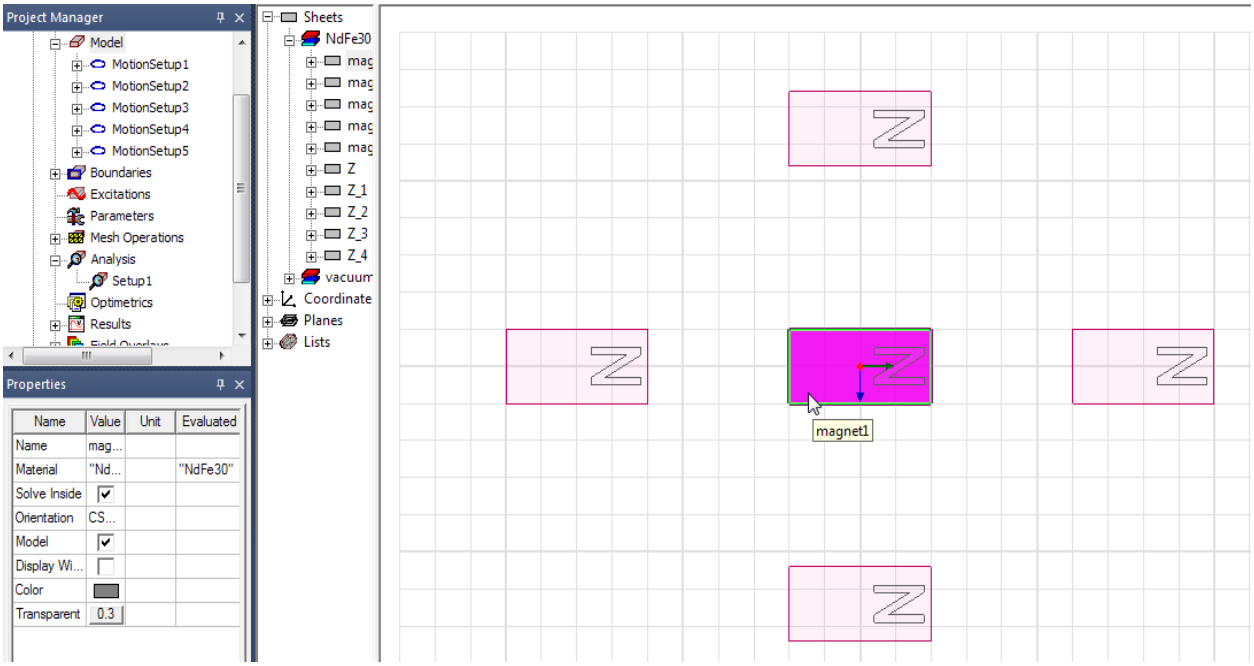
The project shown below comprises two rectangular magnets, each surrounded by a band object with a MotionSetup assigned to it. When one magnet moves, the other magnet realigns itself. Also note that the solution context display shows the position and speed for both motion setups.



Note	The two moving bands may not touch each other. There must be some stationary region between the two bands, even if it is filled only with vacuum.

Example - Multiple Bands of Motion

The magnet project shown below comprises five bar magnets. When the center magnet (MotionSetup1) moves, the other magnets also move.



Note	The moving bands may not touch each other. There must be some stationary region between the bands, even if it is filled only with vacuum.

Related Topics

Assigning a Band of Motion

Showing and Hiding Motion Objects

Motion Variables Tab

Technical Notes: 3D Transient

Technical Notes: Meshing Aspects for 3D Transient Applications With Motion

Assigning a Band of Motion

You can assign a band of motion in both Maxwell 2D and Maxwell 3D projects. For both Maxwell 2D and 3D projects, two or more motion bands may be assigned.

Note	Overlapping bands are not supported so they cannot be located inside of each other.

For both rotational and translational problems, the band object must always enclose all of the moving objects. For rotational problems, this means the band object should be a solid cylinder or a solid wedge – not a hollow shell or segment of a shell. If the band object encloses all moving aspects, it is easier to identify the moving objects and reduces errors related to misidentifying them.

Note	An inner region (or wrapper object) enclosed by the band can normally be used as a "fake" moving object for two reasons:

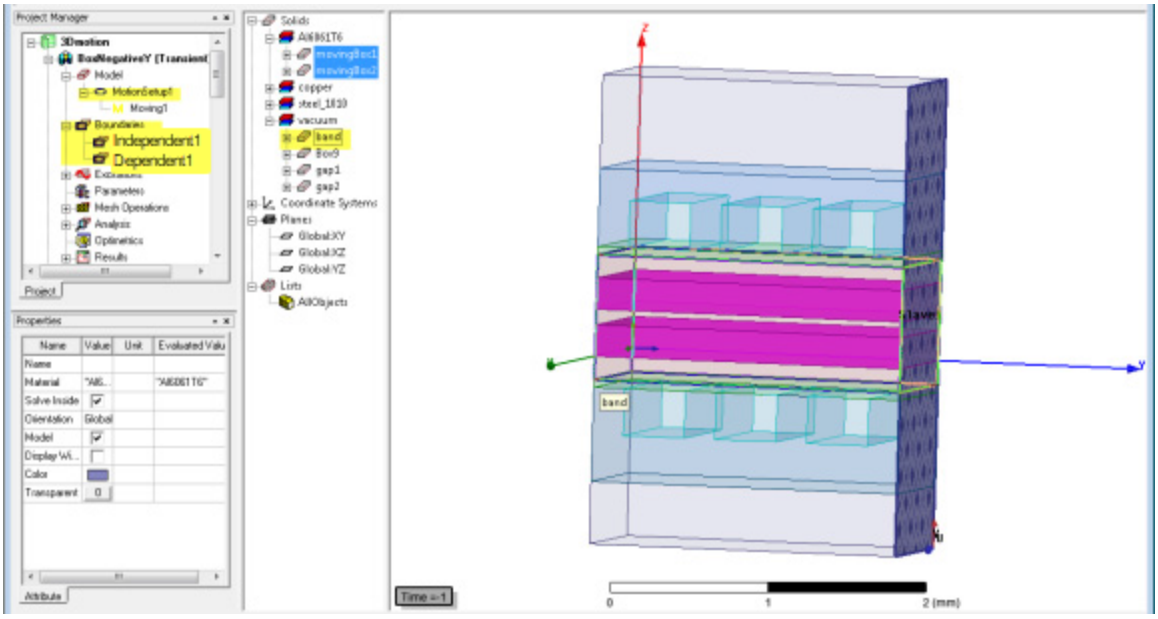
	• The first is to avoid creating holes inside the band which would result in the band having multiple connected domains. • The second is due to the torque/force computation being carried out by one layer of elements inside the band region that directly touches moving objects. If an inner region is not included, the torque/force computation will be very sensitive to mesh noise - thus leading to an inaccurate solution. For 2D and 3D translational motion, the following are some additional recommendations: • An all-encompassing inner region (non-magnetic, non-conductive wrapper such as air or vacuum) must be used around all moving objects in a band. • The inner region wrapper object, like the Band itself, should be a simplified shape, such as a rectangle or polygon in 2D, or a polyhedral or segmented sweep section in 3D. Any simplifications to the inner region and band object will result in better re-meshing for each time-step and faster overall simulation times. • In order to give more accurate forces, if there is sufficient clearance between moving and stationary objects, you can provide some padding between the inner region and the moving objects and the band. At a minimum, the inner region should be the same size as the outside of the enclosed objects (like shrink wrap) with no gap between the wrapper and objects inside. • Ansys highly recommends that the band not touch any outer stationary objects, with the only exception being for symmetry boundaries on the outer region. It is suggested to locate the band halfway in the air-gap between fixed and moving objects or equally spaced with the inner region object if there is sufficient clearance at the gap. • In order to get stable force results, length-based mesh operations (approximately equal to the distance between the band and the inner region) should be applied using the Length-Based-Inside mesh refinement assigned to the Band object so that the mesh length is refined in the band region that is re-meshed for each time-step.

To assign the band of motion:

1. Select the band object.
2. Click **Maxwell>Model>Motion Setup>Assign Band**.

   The **Motion Setup** window appears.
3. On the **Type** tab: select either **Translation** or **Rotation** as the **Motion Type**.
   - For **Rotation** about a fixed point you may select the **Non-Cylindrical** check box.
   - For **Translation**, you may select the **Periodic** check box.

   For **Periodic** motion, you must define independent and dependent boundary conditions. Both band and the object(s) inside it move, and both should touch the independent and dependent boundaries as shown in the following example.



4. Set the motion direction:
   a. For **Translation** (including **Periodic**), select the desired X, Y, or Z (for 3D) axis of motion from the **Moving Vector** pull-down list; then select either **Positive** or **Negative** direction.
   b. For **Rotation**, from the **Moving Axis** pull-down list, select an axis of rotation. Use the radio buttons to select either **Positive** or **Negative** rotation.

Note	• For 2D XY rotation, only **Positive** rotation is allowed. • If the expected moving vector does not exist, click **Modeler>Coordinate System>Create>Relative CS** to create a new coordinate system.

5. On the **Data** tab: type a value in the **Initial Position** box, and select the units from the pull-down list.
6. To set the motion limits, do the following in the **Translate Limit** or **Rotate Limit** section:

Note	For Periodic motion, you do not need to define motion limits because the solver does not consider the limits for periodic motion; thus the **Translate Limit** section is not present.

   a. For rotational motion, click the **Rotate Limit** check box.
   b. For translational (non-**Periodic**) or rotational motion, type a value in the **Negative** box, and select the units from the pull-down list.
   c. For translational (non-**Periodic**) or rotational motion, type a value in the **Positive** box, and select the units from the pull-down list.
7. On the **Mechanical** tab, do one of the following to specify the object velocity (dynamic or constant):

Note	When you select **Consider Mechanical Transient**, you are telling the solver to use the calculated force and user-specified mass, damping, and other

	parameters to dynamically determine how the object moves, rather than giving it a constant velocity.

Note	• For Maxwell 2D/3D designs for motor applications, the computed output mechanical power at given constant speed may not reach the desired rated power. Therefore, it is useful to apply constant power as the mechanical load. In Maxwell 2D/3D designs created by RMxprt, the mechanical transient is set up according to the rated mechanical output power, and equivalent damping is added to accelerate the process approaching to the rated output power. This setup is visible when **Consider Mechanical Transient** is checked. • The**Consider Mechanical Transient**setup is only for the sake of getting the operating point with given output power. it is not related to any mechanical load type you may set in RMxprt. • 2D/3D designs generated from RMxprt include the setup data, but **Consider Mechanical Transient** is unchecked initially. To ensure that the setup data is preserved, make sure that you check **Consider Mechanical Transient** before saving the project.

- To dynamically change how the object moves, click the **Consider Mechanical Transient** check box, complete the following fields, and select any corresponding units:
  - **Initial Velocity** (translational) or **Initial Angular Velocity** (rotational)
  - **Mass** (translational) or **Moment of Inertia** (rotational)

    Use the **Calculate** button to have Maxwell calculate the Mass or Moment of Inertia with the following restrictions:
    - For 2D problems, the moving bodies are assumed to be of model-unit thinness.
    - Rotating bodies must be symmetric about the axis of rotation. Partial models (e.g., half a motor with implied symmetry) are not supported.
  - **Damping** (rotational motion) – A value for damping for rotational motion (for electric machines) can be calculated by the windage and friction loss in Watts divided by the speed in rad/sec (at which those losses are defined); this is Damping with resulting units of (W / (rad/sec)^2) or (N*m * sec/rad).
  - **Damping** (linear motion) – A value for damping for linear motion can be calculated by the windage and friction loss in Watts divided by the speed in m/sec (at which those losses are defined); this is Damping with resulting units of (W / (m/sec)^2) or (N*m * sec/m).
  - **Load Force** (translational) or **Load Torque** (rotational)

Note	• For translational motion, the load force is positive if it is applied in the same direction as the moving vector and negative if applied in the opposite direction. • For rotational motion, the load torque sign is determined based on the moving vector, using the right-hand rule. Point your thumb in the direction of the moving vector. If the load torque is applied in the same

	direction as your fingers, then the load torque is positive. If it is applied in the opposite direction, then the load torque is negative.

- To set a constant velocity (and disable the mechanical transient), clear the **Consider Mechanical Transient** check box, type a value in the **Velocity** or **Angular Velocity** box, and select the units from the pull-down list.

8. For translational motion, a **Post-Processing** tab is available to set the coordinate system for post processing calculation.
9. Click **OK**.
   - A **MotionSetup*x*** is created and added to the **Model** folder in the **Project Manager** window.
   - For Maxwell 3D projects only, if cylindrical **Rotation** was specified and a **CylindricalGap** mesh operation does not already exist, a **CylindricalGap** mesh operation is automatically added to the **Mesh** folder. Only one **CylindricalGap** mesh operation is permitted.

Related Topics

Non-Cylindrical Rotational Motion
Showing and Hiding Motion Objects
Reassigning a Band of Motion
Motion Variables Tab
Creating Animations

Modifying Motion Setup Properties

To modify existing motion setup properties:

1. In the Project Manager Model folder, right-click on **MotionSetup*x*** and select **Properties** from the context menu to open the **Motion Setup** dialog box.
   - Alternatively, you can double-click the **MotionSetup*x*** icon.
2. Edit the desired properties in the **Motion Setup** dialog box. Refer to *Assigning a Band of Motion* for detailed information on the various **Motion Setup** dialog box settings.

Note	For Maxwell 3D projects only: • If you change the motion type to cylindrical **Rotation** and a **CylindricalGap** mesh operation does not already exist, a **CylindricalGap** mesh operation is automatically added to the **Mesh** folder. Only one **CylindricalGap** mesh operation is permitted. • If you change a cylindrical **Rotation** motion type to another motion type, the **CylindricalGap** mesh operation remains in the **Mesh Operation** folder.

   - Alternatively, you can edit many of the motion setup properties in the **Properties** window.

Related Topics

Non-Cylindrical Rotational Motion

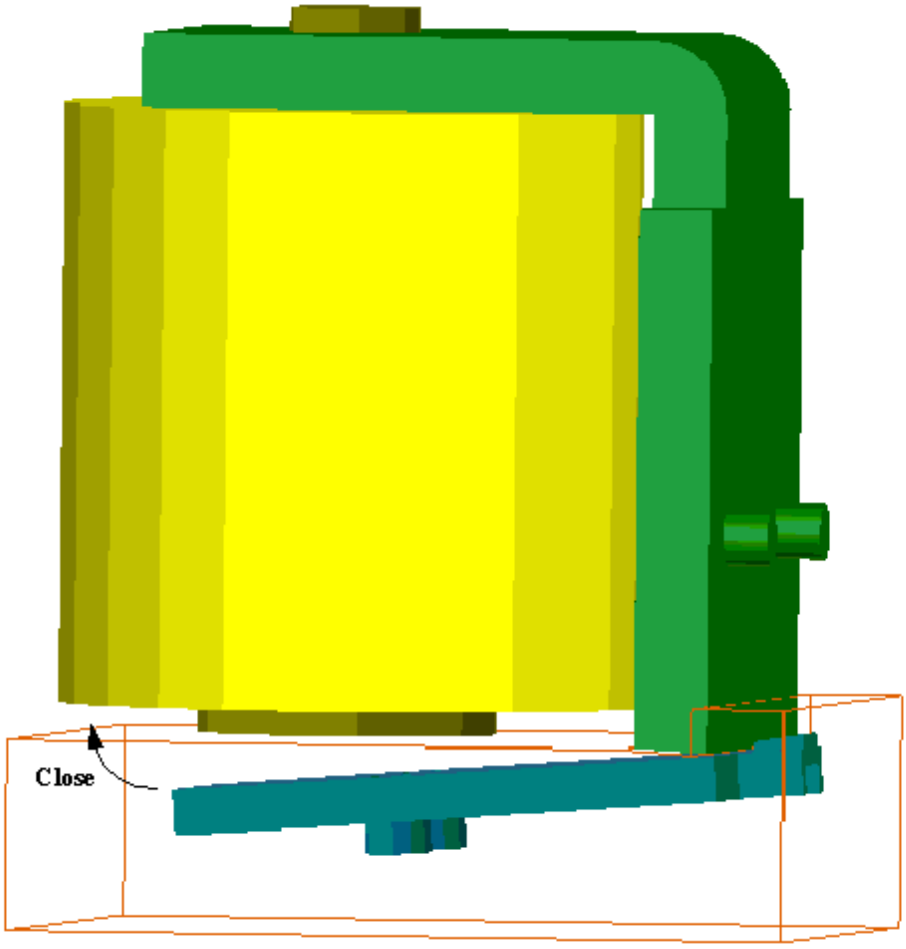
Showing and Hiding Motion Objects

Reassigning a Band of Motion

Non-Cylindrical Rotational Motion

With cylindrical rotation, the moving objects are free to rotate completely, and the mesh remains fixed as the moving objects rotate. With non-cylindrical motion, the geometry shape of the band region changes as the moving objects rotate, and the mesh is recreated.

- A motor is an example of cylindrical motion.
- A relay is an example of non-cylindrical motion.



Reassigning a Band of Motion

When a **MotionSetup** has been created, the object that the setup is assigned to is changed using the **Reassign** command, allowing you to identify the motion setup parameters with a new object

without losing the motion setup parameters themselves. To reassign a **MotionSetup**:

1. Select the object that the **MotionSetup** will be reassigned to in the modeler window or in the history tree.
2. Right-click on **MotionSetup*x*** in the Project Manager window and select **Reassign**.

Note	• An existing **CylindricalGap** mesh operation's assignment will be synced up to a rotational cylindrical motion setup's assignment when you reassign the band object of the motion setup.

Related Topics

Assigning a Band of Motion

Adding and Deleting Objects from a Band of Motion

To add objects to, or delete objects from a motion setup without changing the parameters of the **MotionSetup** do the following:

1. Select the object to be added or deleted from the **MotionSetup** in the modeler window or in the history tree.
2. Right-click **Model>MotionSetup*x*>Moving*x*** in the Project Manager window and choose **Add Selected Objects** or **Remove Selected Objects** from the shortcut menu.
3. **To verify the addition or deletion of the object from the MotionSetup definition, use Select Assignment from the shortcut menu to view the objects assigned to the band.**

Related Topics

Assigning a Band of Motion

Showing and Hiding Motion Objects

1. Click **Maxwell>Model>Motion Setup>Visualization**.
   The **Motion Setup Visualization Options** window appears.
2. Select **All**, **Band**, or **Moving** from the **Select** pull-down list.
3. Click **Deselect All** to hide all motion objects.
4. Click **Close** to close the window.

Related Topics

Setting the Visibility for Motion Objects

Setting the Visibility for Motion Objects

To show or hide a parameter:

1. Click **View>Active View Visibility**.
   The **Active View Visibility** dialog box appears.
2. Click the **Motion** tab.
3. Select or clear the **Visibility** check box for each motion object listed.
4. Click **Done**.

Modeling Maxwell Motion Eccentricity

You can model motion eccentricity for Maxwell 2D and 3D transient designs that have cylindrical rotational motion set up. Ansys Electronics Desktop provides a **Maxwell Eccentricity Wizard** with settings for:

- Rotating part eccentricity
- Rotation axis eccentricity

For rotating part eccentricity modeling, the eccentricity is only applied to rotating objects, not to a motion axis. More specifically, these rotating objects are all objects enclosed by the band, but not the band itself.

For rotation axis modeling, the eccentricity is only applied to the motion axis, which is equivalent to moving the band itself, but not any rotating objects inside the band.

Using these two basic settings, you can model the following physics.

- **Static eccentricity**: To model static eccentricity, you need to set the rotating part eccentricity and rotation axis eccentricity to the same values.
- **Dynamic eccentricity**: To model dynamic eccentricity, set the rotating part eccentricity values only.
- **Mixed eccentricity**: This is the combination of static and dynamic eccentricities. To model mixed eccentricity, set both rotating part eccentricity and rotation axis eccentricity, but with different values.

Related Topics

Using the Maxwell Eccentricity Wizard
Working with the Extensions Window
Non-Cylindrical Rotational Motion
Showing and Hiding Motion Objects
Reassigning a Band of Motion
Motion Variables Tab
Creating Animations

14 - Adding Solution Parameters

When you assign a solution parameter, the solver calculates the parameter value as part of the solution process. The table below shows the solution parameters that may be calculated based upon the solver type.

Parameter	3D Solvers	2D Solvers
Force/Torque	Magnetostatic Electrostatic Eddy Current Transient Electric Transient	Magnetostatic Electrostatic Eddy Current Transient
Matrix	Magnetostatic Electrostatic Eddy Current DC Conduction AC Conduction	Magnetostatic Electrostatic Eddy Current AC Conduction DC Conduction

Assigning a Force Parameter

To solve for a force parameter:

1. Select the object on which you want to apply the parameter.
2. Click **Maxwell3D or Maxwell2D, and then select Parameters>Assign>Force**. You can also right-click on Parameters in the Project tree, and then select **Assign>Force** from the context menu.

   The **Force Setup** window appears.
3. Type a name for the force in the **Name** box.
4. For **Maxwell3D** projects only, select **Virtual** or **Lorentz** as the **Type** for the force.

   > **Note:**
   >
   > - For 3D Magnetostatic, Electrostatic, Eddy Current, and Transient solvers, you can select either **Virtual** or **Lorentz** force to be calculated.
   > - For 3D Electric Transient solvers, only **Virtual** force is calculated.
5. Select the **Post-Processing** tab to specify the coordinate system reference for calculating the force parameter.

> **Note:** If object or face coordinate systems are used as reference in the post processing tab of the force setup in a transient analysis with motion, it should be noted that the original position of the CS will be used for this coordinate transformation. This coordinate transformation will not consider the motion enforced by the motion setup.

6. Select the appropriate coordinate system from the **ReferenceCS** pull-down list.
7. Click **OK**.

> **Note:** Force parameters are available for calculation in the 2D and 3D Magnetostatic, Electrostatic, Eddy Current and Transient solvers, and for the 3D Electric Transient solver only.

> **Note:** The calculation of forces using the A-Phi formulation requires an extremely fine mesh, since B is solved as 0th order and is constant at each node of each tetrahedral element.

Assigning a Torque Parameter

To solve for a torque parameter:

1. Select the object on which you want to apply the parameter.
2. Click **Maxwell3D or Maxwell2D, and then select Parameters>Assign>Torque**. You can also right-click on Parameters in the Project tree, and then select **Assign>Torque** from the context menu.

   The **Torque** setup window appears.
3. Type a name for the torque in the **Name** box.
4. Select **Virtual** or **Lorentz** as the **Type** for the torque.

Note	For 3D Transient and Electric Transient solvers, only **VirtualTorque** is calculated.

5. To assign an axis:
   a. Select the axis from the **Axis** pull-down list.
   b. Select either the **Positive** or **Negative** radio button.
6. Click **OK**.

Note	Torque parameters are available for calculation in the 2D and 3D Magnetostatic, Electrostatic, Eddy Current, and Transient solvers, and for the 3D Electric Transient solver only.

Assigning a Matrix

To solve for a capacitance, inductance, impedance, or conductance matrix:

1. Click **Maxwell 3D** or **Maxwell 2D**, and then select **Parameters>Assign>Matrix**.

   The **Matrix** dialog box appears.
2. Click the **Setup** tab.
3. Type a name for the matrix in the **Name** box.
4. To specify the sources to be included in the matrix, do one of the following:
   - For 3D Electrostatic, 2D and 3D Magnetostatic, 2D and 3D Eddy Current designs, select or clear the **Include** check box for any of the listed sources.

     For **Maxwell 2D** designs, the return path for each source may be specified. By default, the return path is at infinity; however, any conductor with a source specified can act as the return path.
   - For 2D Electrostatic, 2D and 3D DC Conduction, and 2D and 3D AC Conduction, select or clear the **Signal** or **Ground** check box for any of the listed sources. Grounded terminals are treated as reference voltages with zero voltage values.

     Excitations checked as **Signal** will be in the matrix (excited one at a time with 1V during the parameters extraction process). Those excitations checked as **Ground** will be kept at 0 V and will not be part of the matrix. Any excitations not checked will not be considered.
5. For Magnetostatic designs, click the **Post Processing** tab.

   If more than one source is listed on the left side, you can click the **Post Processing** tab and **Group** them to organize multiple sources into a single group. The calculations defined in the Post Processing tab do not impact the field solution and can be set up or changed after the solution has been completed.

   To **Group** multiple sources together:
   a. If desired, set the number of **Turns** for each source.
   b. Select the sources to group in the list **Entry** list. Multiple source can be selected by holding the **CTRL** key and clicking each source line.
   c. With all sources selected, click **Group** to combine the sources into one series winding.
   d. Set the number of parallel **Branches** the winding should be divided into and click **OK**.
   e. The results of the post processing calculation on the matrix can be seen by selecting **PostProcessed** on the **Matrix** tab of the solutions dialog box.
6. Click **OK**.

Note	• For a matrix setup, the elements are defined based on excitations, not solids. Only excitations will be available in the **Matrix** dialog box, regardless of the current geometry selection. Also, changes made to the excitations after matrix setup will not be flagged until the setup validation performed prior to the solution process.

	• Matrix setups are recognized when coupling Maxwell designs to Twin Builder components such as Equivalent Circuit, Dynamic Inductance and Capacitance, and State Space. For more information on coupling Maxwell designs to Twin Builder components, refer to the *Maxwell Component Subcircuits* topic in the Twin Builder Help. • For 2D and 3D Eddy Current designs, you can also *Assign a Reduce Matrix*. • For the 3D AC Conduction solver, the current excitations might also be considered during the matrix extraction. They will be considered in the same way as the regular voltage excitations.

In a matrix calculation:

- For coil terminals, all terminals are listed and can be selected.
- For outer terminals, the solver can only handle a conduction path with two terminals, and only one terminal is listed for each conduction path:
  - If they are current terminals, the listed terminal is in the “in” direction.
  - If they are current density terminals, the first terminal is listed and can be selected.
- When setting up an inductance matrix, the source must be a current excitation. Voltage sources cannot be used in this case.
- In 2D matrix calculations the solver calculates loop currents; therefore, the direction of return path current is always opposite of the source current. If a conductor with excitation is specified as the return path, the polarity setting of the excitation in the nominal problem setup is not respected.

Related Topics

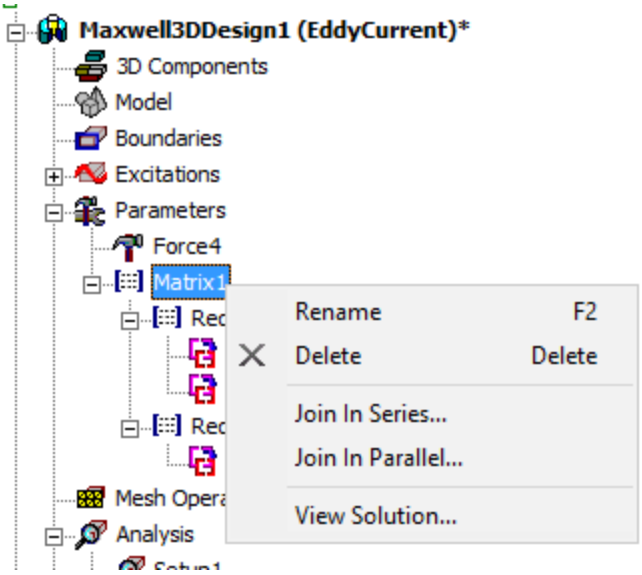
Assigning a Reduce Matrix
Viewing Matrix Data
Capacitance Matrix for an Electric Field Solution
Inductance Matrix for a Magnetostatic Field Solution
Conductance Matrix for a 3D DC Conduction Solution
Grouping Inductance Matrix Elements
Impedance Matrix for an Eddy Current Solution
Post-Processed Quantities

Assigning a Reduce Matrix for 2D and 3D Eddy Current Designs

For 2D and 3D Eddy Current designs, you can join (group) two or more excitations to one excitation in either a series or a parallel connection. The result of each join (group) operation is known as a Reduce Matrix. This would be useful when analyzing designs such as a motor with 4 poles, each pole having 3 wires coming into it, with the physical design having each set of three wires in series, and the four bundles of wires in parallel.

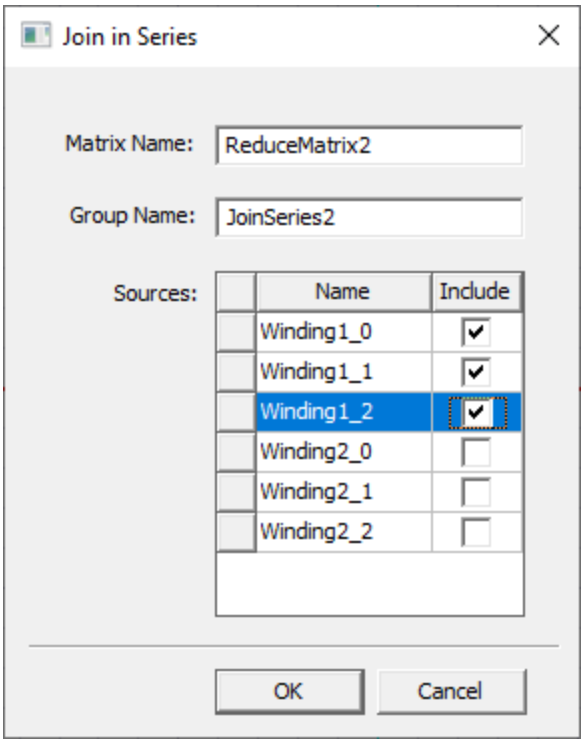
To create a reduce matrix:

1. Right-click on an existing Matrix parameter in the project tree.



2. Select either **Join in Series** or **Join in Parallel**.

   A corresponding **Join in Series** or **Join in Parallel** dialog box opens. You can accept the default reduce **Matrix Name**, or enter the desired name.



3. You can enter a **Group Name** (or accept the default), and check **Include** for those sources you wish to join in series (or parallel).

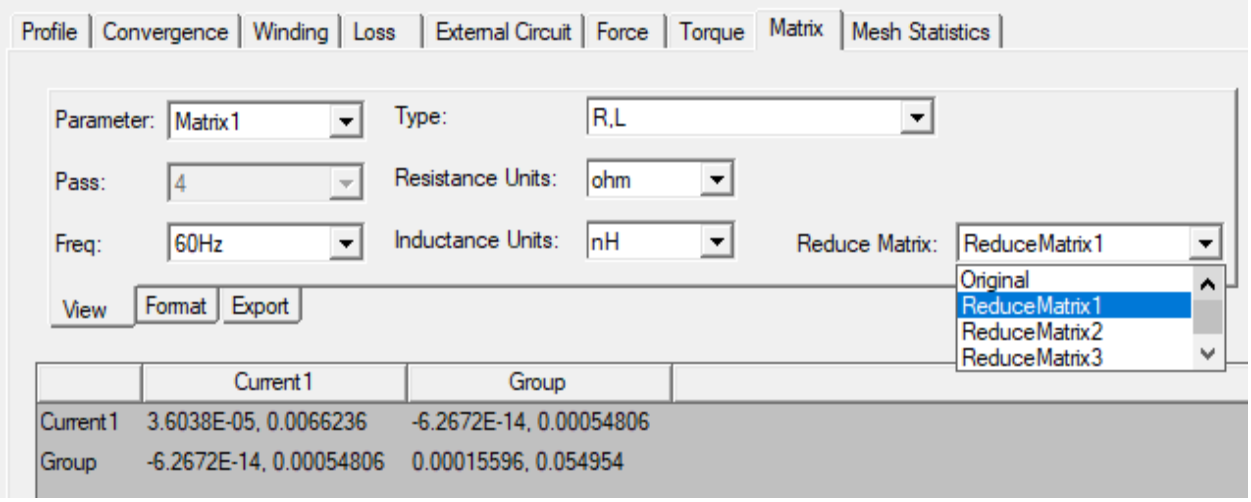
Note	Multiple group operations are supported. For example, you could join conductors 1, 2, and 3 in one series; and conductors 4, 5, and 6 in another series.

4. Click **OK** to close the dialog and add the new reduce matrix group under its parent Matrix.

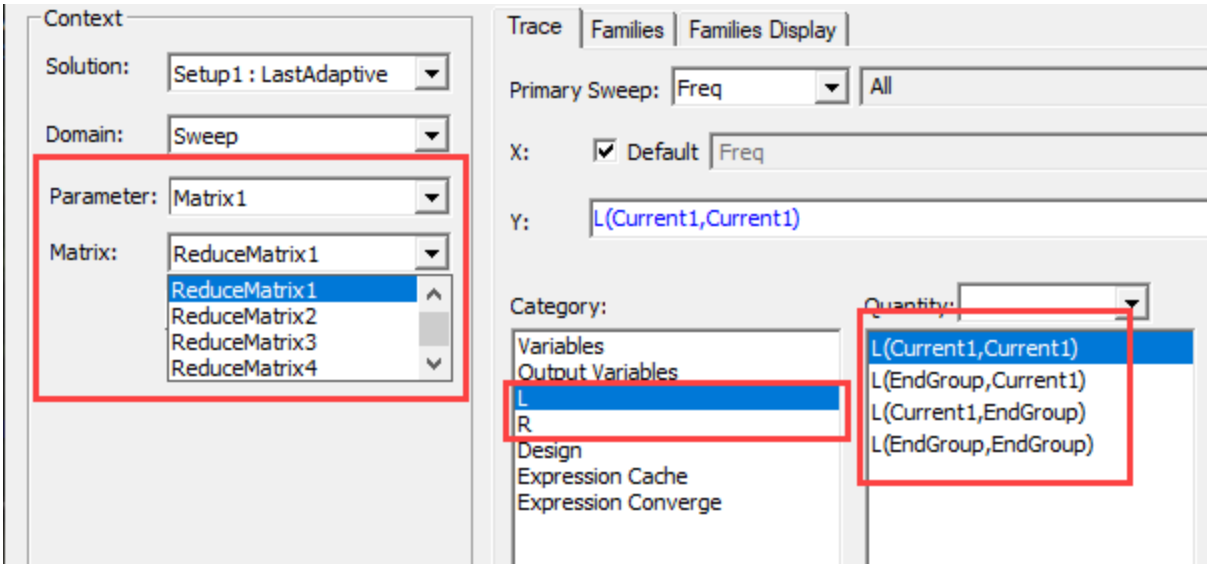
Note	If you want to make changes to a reduce matrix, you can double-click on the reduce matrix group item in the tree to reopen the dialog box. You can also right-click the item to rename or delete it.

If you want to make changes, you can double-click on the reduce matrix group item in the tree to reopen the dialog box. You can also right-click the item to rename or delete it.

You can view the results of reduce matrix calculations on the Matrix tab of the Solutions dialog.



You can select reduce matrices to plot inductance and resistance quantities for various report types.



Related Topics

Viewing Matrix Data

Creating a New Report

Reassigning Parameters

To reassign the geometry a parameter applies to:

1. Select the new object to which you want to assign an existing parameter.
2. Click **Maxwell3D or Maxwell2D, and then select Parameters>Reassign.**
   The **Reassign Parameter** window appears.
3. Select the force or torque you want to reassign from the **Reassign Geometry for** list.
4. Click **OK**.
   The parameter is reassigned to the new object.

Viewing Parameter List

To view a list of all parameters in the design:

1. Click **Maxwell3D** or **Maxwell2D,** and then select **Parameters>List**.
   The **Design List** window appears, with the **Parameters** tab visible.
2. To view other design choices, click one of the following tabs:
   - Model
   - Boundaries
   - Excitations

- Mesh Operations
- Analysis Setup

Deleting All Parameters

To delete all parameters assigned in the model:

- Click **Maxwell3D or Maxwell2D, and then select Parameters>Delete All**.
  All parameters are deleted.

Warning	Maxwell provides no warning or option to stop deletion. If you accidentally delete all parameters, click **Edit>Undo Delete All Parameters** before you take any other action.

Showing and Hiding Solution Parameters

1. Click **Maxwell3D or Maxwell2D, and then select Parameters>Visualization**.
   The **Parameter Visualization Options** window appears.
2. Select the type of parameter you want to view from the **Select** pull-down list, or click the row in the table. Choices include **All**, **All Forces,All Torques**, **All Matrices**, and **By Name**.
   The object is highlighted in the geometry.
3. Select or clear the check boxes for each parameter to show or hide geometry, name, and vectors for each force or torque parameter.
4. Click **Deselect All** to hide all parameters.
5. Click **Close** to close the window.

Setting the Visibility for Parameters

To show or hide a parameter:

1. Click **View>Active View Visibility**.
   The **Active View Visibility** dialog box appears.
2. Click the **Parameters** tab.
3. Select or clear the **Visibility** check box for each parameter listed.
4. Click **Done**.

Changing the Parameter Setup

To change a parameter's setup:

1. In the project tree, open the **Parameters** section.
2. Right-click the parameter you want to change.
   A shortcut menu appears.

3. Select **Properties** from the shortcut menu.
4. Make the desired changes, and then click **Close**.

15 - Defining Mesh Operations

In Maxwell, mesh operations are optional mesh refinement settings that provide Maxwell with mesh construction guidance. This technique of guiding Maxwell's mesh construction is referred to as "seeding" the mesh. Seeding is performed using the **Mesh** commands on the **Maxwell3D or Maxwell2D** menu.

When defining a mesh, you typically assign the mesh operations first (using the **Maxwell3D>Mesh** or **Maxwell2D>Mesh** commands) and then create the mesh (using the **Maxwell3D** or **Maxwell2D** menu and selecting **Analysis Setup>Generate Mesh** command). However, you can also refine the mesh after the initial mesh has been created.

You can instruct Maxwell to refine the length of tetrahedral elements on a surface or within a volume until they are below a certain value (length-based mesh refinement) or you can instruct Maxwell to refine the surface triangle length of all tetrahedral elements on a surface or volume to within a specified value (skin depth-based mesh refinement). In some circumstances, you may also want to create a mesh operation that modifies Maxwell's surface approximation settings for one or more faces. In some circumstances, you may also want to assign a cylindrical gap treatment mesh operation. For Maxwell 2D, when the TAU mesher is selected in the Initial Mesh Settings dialog box, you can define the Skin Depth Layers mesh setting corresponding to the selected edges.

Surface approximation settings are only applied to the *initial mesh* (the mesh that is generated the first time a design variation is solved). The other types of mesh operations (refining the mesh) can be performed on either the initial mesh or on the most recently generated mesh (the *current mesh*) if you are updating a previously-created mesh.

Note	Mesh operations at the target design of a mu (permeability) link are ignored.

You can also choose to override automatic choice of which mesher Maxwell uses, by using the Maxwell>Mesh>Initial Mesh Settings command.

What do you want to do?

- Perform length-based mesh refinement on object faces.
- Perform length-based mesh refinement inside objects.
- Perform skin depth-based mesh refinement on object faces.
- Assigning Cylindrical Gap Treatment
- Apply Skin Depth Layers Settings
- Modify surface approximation settings for one or more faces.
- Modifying the Model Resolution (Maxwell3D only)
- Apply mesh operations
- Specify the initial mesh settings

Related Topics

Generating the Mesh Without Solving

Monitoring the Solution Process

Viewing Mesh Statistics

Plotting the Mesh

Technical Notes: Meshing Aspects for 3D Transient Applications With Motion

Technical Notes: The Mesh Generation Process

Technical Notes:Seeding the Mesh

Technical Notes:Guidelines for Seeding the Mesh

Technical Notes: Surface Approximation Settings

Assigning Length-Based Mesh Refinement on Object Faces

1. Select the faces you want Maxwell to refine.

   Alternatively, select an object if you want Maxwell to refine every face on the object.
2. Click **Maxwell>Mesh>Assign Mesh Operation>On Selection>Length Based**.

   The **Element Length BasedRefinement** dialog box appears.
3. Type a name for the mesh operation in the **Name** box, or accept the default name.
4. To restrict the length of tetrahedra edges touching the faces:
   a. Select **Restrict Length of Elements**.
   b. Type the maximum length of the tetrahedral edges touching the faces in the **Maximum Length of Elements** box, and select the units.

Note	Maxwell refines the element edges touching the selected faces until their lengths are equal to or less than this value. The default value is set to 20% of the maximum edge lengths of the bounding boxes of each selected face.
5. To restrict the number of elements added during refinement of the faces:
   a. Select **Restrict the Number of Elements**.
   b. Enter the **Maximum Number of Elements** to be added.
6. Click **OK**.

When the mesh is generated, the refinement criteria you specified are used. When the maximum number of elements is reached, some elements may exceed the requested maximum element length.

You can also specify Initial Mesh Settings to apply to all objects; however, if you apply separate surface approximation mesh operations to specific objects, the object settings take precedence over the general setting.

Related Topics

Assigning Length-Based Mesh Refinement Inside Objects

Generating the Mesh without Solving

Plotting the Mesh

Technical Notes:Length-Based Mesh Refinement

Technical Notes: The Mesh Generation Process

Technical Notes:Seeding the Mesh

Technical Notes:Guidelines for Seeding the Mesh

Assigning Length-Based Mesh Refinement Inside Objects

Length-based mesh operations inside volumes can be applied to non-model box primitives. If an error occurs, you may be trying to assign a mesh to a non-model box that is not a box primitive (such as boxes with chamfers or non-model cylinders).

To instruct Maxwell to refine every face of an object and its interior:

1. Select the object you want Maxwell to refine.
2. Click **Maxwell>Mesh>Assign Mesh Operation>Inside Selection>Length Based**.

   The **Element Length Based Refinement** dialog box appears.
3. Type a name for the mesh operation in the **Name** box, or accept the default name.
4. To restrict the length of the tetrahedral element edges inside the object:
   a. Select **Restrict Length of Elements**.
   b. Type the maximum length of the edges inside the object in the **Maximum Length of Elements** box, and select the units.

Note	Maxwell refines the element edges inside the object until they are equal to or less than this value. The default value is set to 20% of the maximum edge lengths of the bounding boxes of each selected object's faces.
5. To restrict the number of elements added during the refinement inside the object:
   a. Select **Restrict the Number of Elements**.
   b. Enter the **Maximum Number of Elements** to be added.
6. Click **OK**.

When the mesh is generated, the refinement criteria you specified are used. When the maximum number of elements is reached, some elements may exceed the requested maximum element length.

You can also specify Initial Mesh Settings to apply to all objects; however, if you apply separate surface approximation mesh operations to specific objects, the object settings take precedence over the general setting.

Related Topics

Assigning Length-Based Mesh Refinement on Object Faces

Generating the Mesh without Solving

Plotting the Mesh

Technical Notes:Length-Based Mesh Refinement

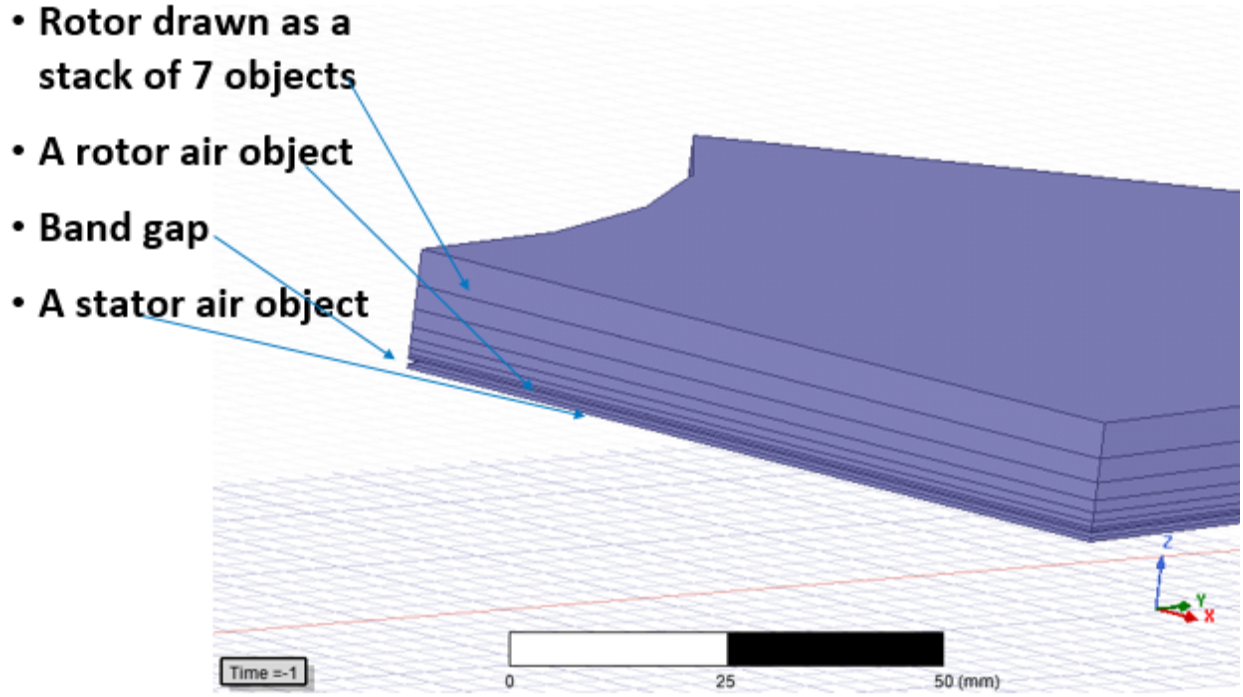
Technical Notes: The Mesh Generation Process

Technical Notes:Seeding the Mesh

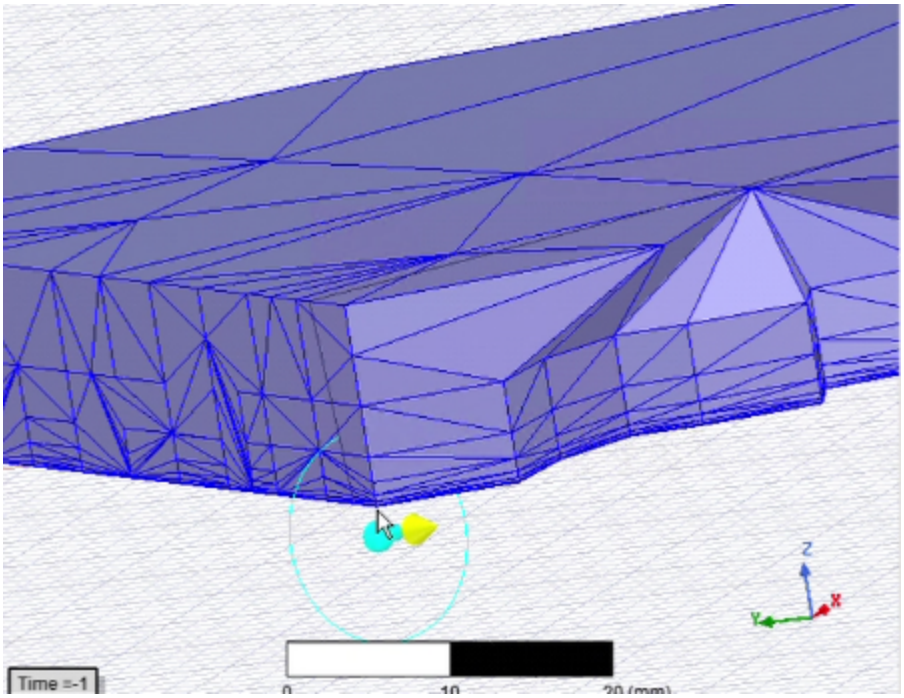
Technical Notes:Guidelines for Seeding the Mesh

Assigning Skin Depth-Based Mesh Refinement on Object Faces

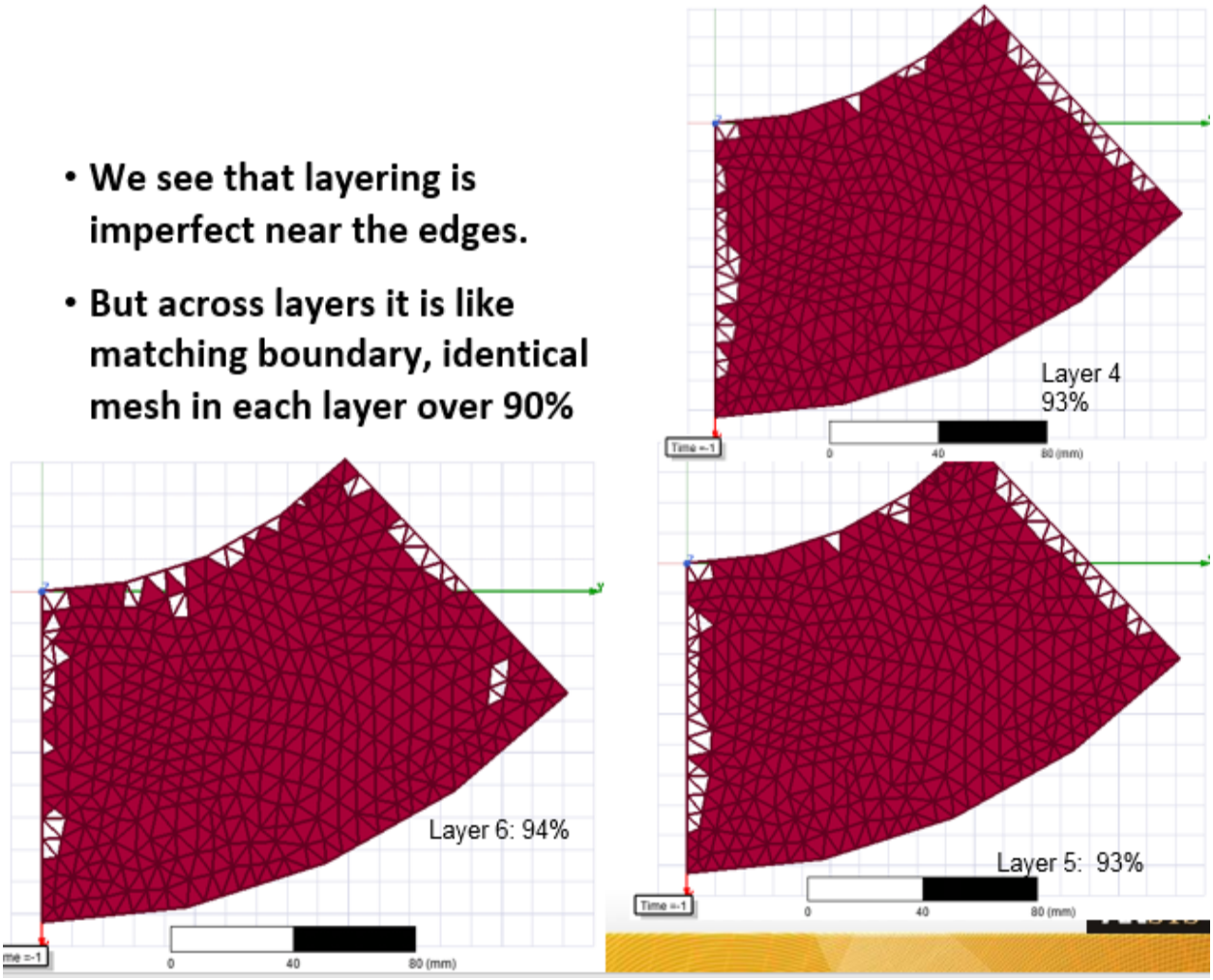
The Skin Depth-based mesh refinement lets you calculate or specify a skin-depth for mesh refinement. You can also specify the number of layers of elements for refinement where the skin depth is the total depth of all layers combined. These layers provide an easy, alternative approach than by creating physical models of each layer, pseudo-sheet bodies. Whereas creating and adjusting a complex, layered physical model is difficult, changing the skin depth parameters and number of layers is very easy. For example, consider a Maxwell rotor, drawn as a stack of seven objects:



To test various stacking models with pseudo bodies involves recreating the model with different thicknesses and remaking the entire initial mesh. The Tau mesher may not be available for all layer heights. You can achieve comparable results by using skin depth based refinement by reverting to the initial mesh, changing settings and applying mesh operations. The following figure shows a cut-plane image of the interior of a model that uses skin-depth based refinement and layers of elements. The Skin Depth is the total depth of all layers combined. Layered elements apply to the selected faces of solid bodies. The elements are stretched parallel to face, and are compressed in the normal direction.



While in some places the edges have imperfect layering, the interior is very good. You can also view the results layer by layer by using the Model Analysis dialog.



The layered skin depth mesh operation provides a tool toward finding a good solution. You can try different settings to improve results toward a good solution. For instance, for this example, these different settings were tried. For comparison, a model built with seven layers required 85K tets.

Tri length target mm	Total skin depth mm	Number of layers	Mesh Size element count	Perfect layering by tri count	Perfect layering by area
15.0	6.3	6	63K	43%	65%
10.0	6.3	6	68K	95%	94%
8.0	6.3	6	76K	96%	93%
6	6.3	6	97K	97%	94%

The goal is not to have perfect layering, but to have a good solution that is much easier to achieve.

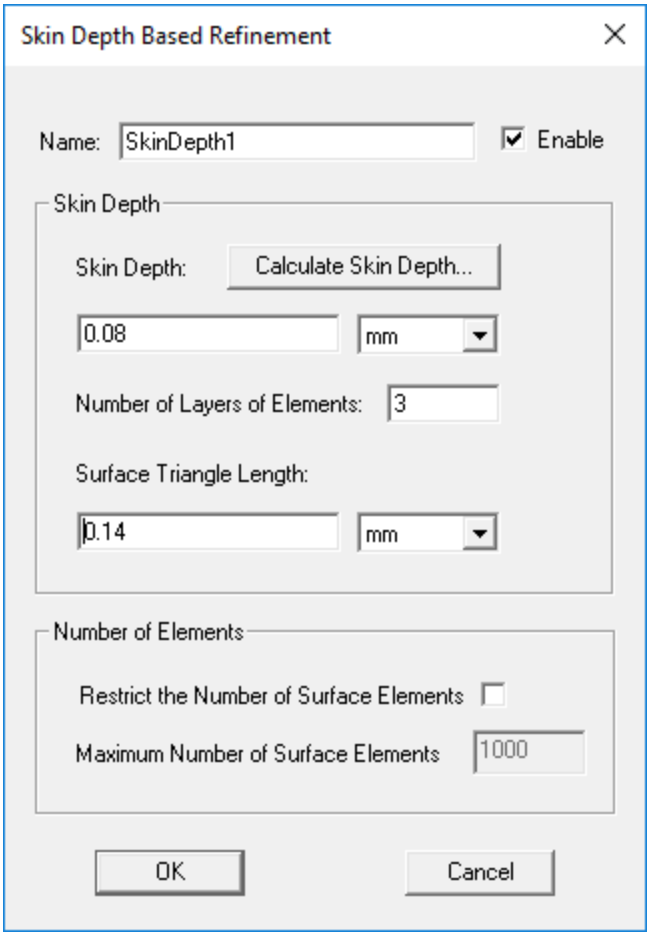
To use Skin Depth-Based Refinement:

1. Select the faces you want to be refined.

Note	It is possible to select a body and convert it to selecting all faces of the body. The user can use this method to select all faces and toggle a few faces out of selection. Selecting the whole body might select very large regions for refinement and increase the element count a lot.

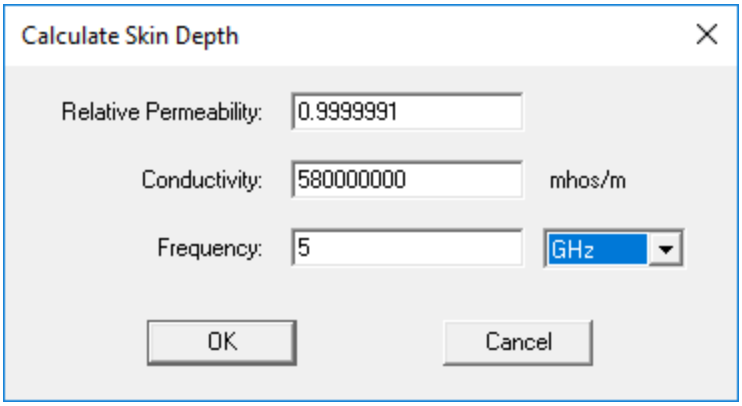
2. Click **Maxwell>Mesh>Assign Mesh Operation>On Selection>Skin-Depth-Based**.

   The **Skin Depth-Based Refinement** dialog box appears.



3. To calculate the skin depth based on the object's material permeability and conductivity and the frequency at which the mesh will be refined, click **Calculate Skin Depth**.

The **Calculate Skin Depth** dialog box appears with values based on the selected object's material properties and the solution setup frequency.



Accept or edit these values. When you click **OK** the solver calculates the skin depth and enters its value in the **Skin Depth** text box. You can accept the calculated values or provide your own.

4. In the **Number of Layers of Elements** text box, type the number of layers to add perpendicular to the object's surface. The skin depth is the total depth of all layers combined.

   The solver will add an equivalent number of mesh points to each layer. For example, if Maxwell added 10 points to satisfy the **Surface Triangle Length**, it will add 10 points to each layer.

5. Optionally, provide the maximum edge length of the surface mesh in the **Surface Triangle Length** text box. The default value is set to 20% of the maximum edge lengths of the bounding boxes of each selected face.

   The solver will refine the surface triangle mesh (the faces of the tetrahedra touching the surface) until their edge lengths are less than or equal to the specified value.

6. By default, the **Restrict Number of Surface Elements** setting is unchecked. This allows the mesher to use symmetry more effectively. If you Restrict the number of surface elements, this may affect symmetry but can be used to protect against runaway refinement in specific cases. To restrict the number of elements added during refinement on the faces uncheck the box to enable the field for **Maximum Number of Surface Elements**. With the box unchecked, the field, grayed out, the number in the box provides an estimate of mesh growth.

When the mesh is generated, the refinement criteria you specify are used. This operation will be approximately the same as having slabs of tetrahedra, but it is not guaranteed to prevent tetrahedra from crossing slab interfaces. Caution should be used with this mesh operation, as very thin layers may cause a reduction in mesh quality or unnecessarily cause the generation of a very large mesh. Further regions refined under this operation and its close neighbors do not participate in solution adaptive refinement. This is another reason to use this seeding operation with caution.

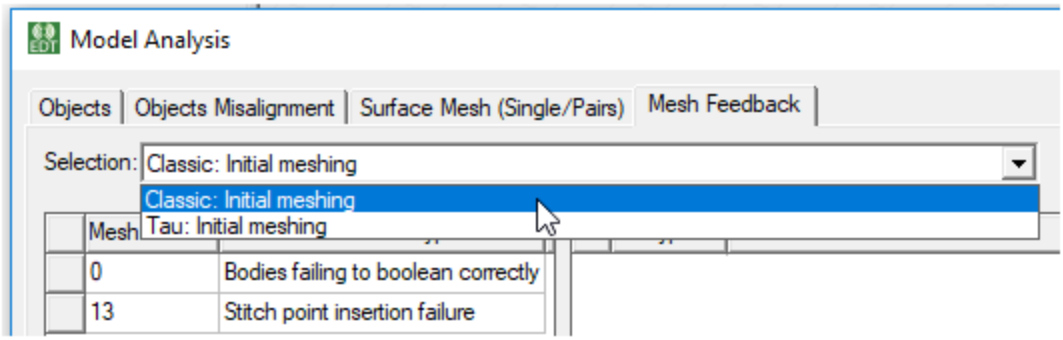
You can also specify Initial Mesh Settings to apply to all objects; however, if you apply separate surface approximation mesh operations to specific objects, the object settings take precedence over the general setting.

Visualization for Skin Depth Refinement Mesh Results

You can visualize the mesh to assess results either by using the clip-plane feature on a model with a mesh plot, or by using the Model Analysis dialog as follows:

1. Select the object on which you have generated a meshplot, and click **Modeler>Model Analysis>Show Analysis Dialog>View Mesh Feedback**.

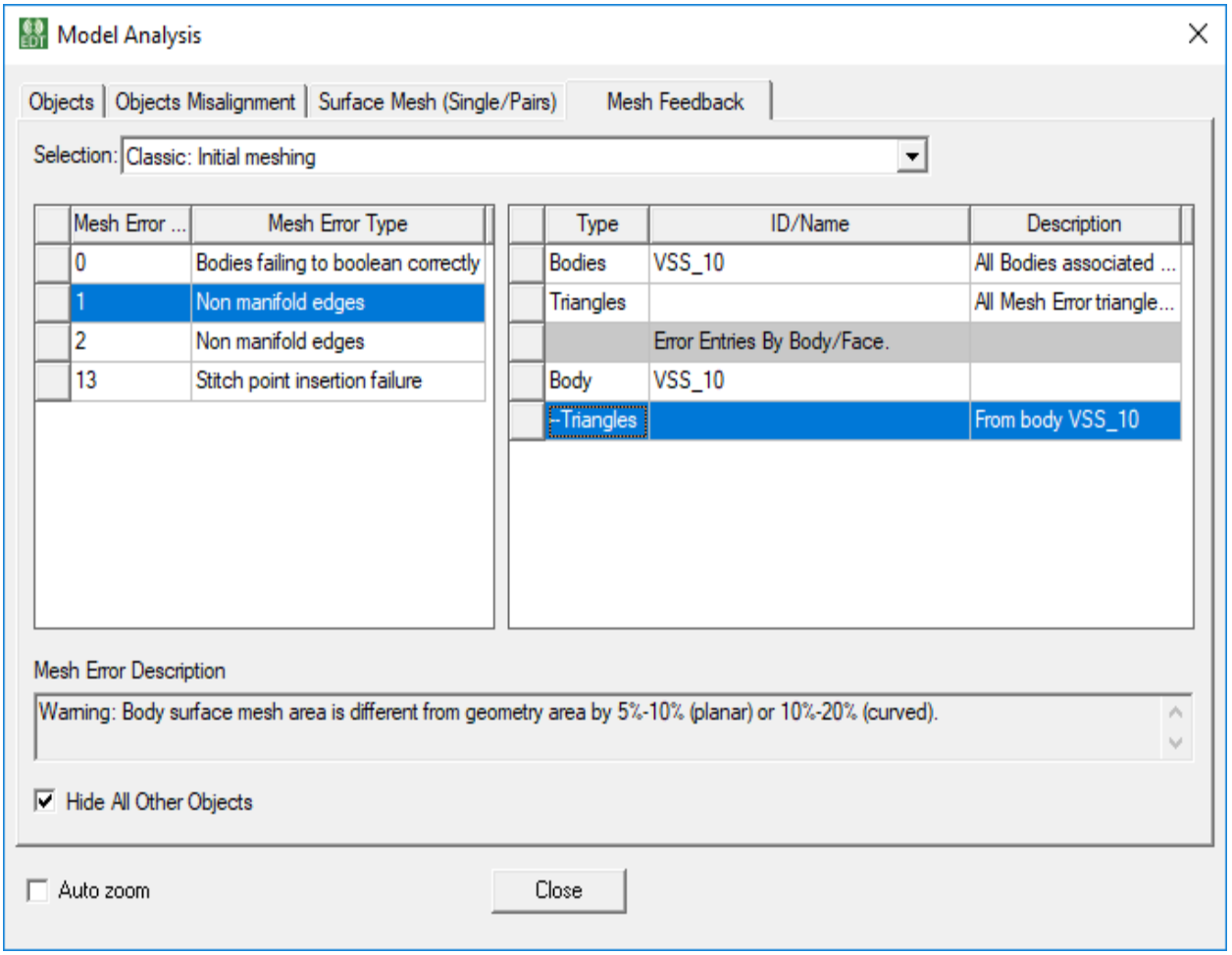
   The **Model Analysis** dialog appears. If the project has solutions by different solvers, the **Selection** drop down lets yous select which mesh to view.



2. On the upper left table, under Mesh Error, select 0 through n , where 0 is the selected body, 1 is the layer closest to the face, 2 is the next, up to layer "n" as farthest.

   Selecting one of these rows causes a **Mesh Error Description** to display for the selected layer or body. The message gives the Total Expected count for triangles, as

compared with the Success percentage by count and by area.



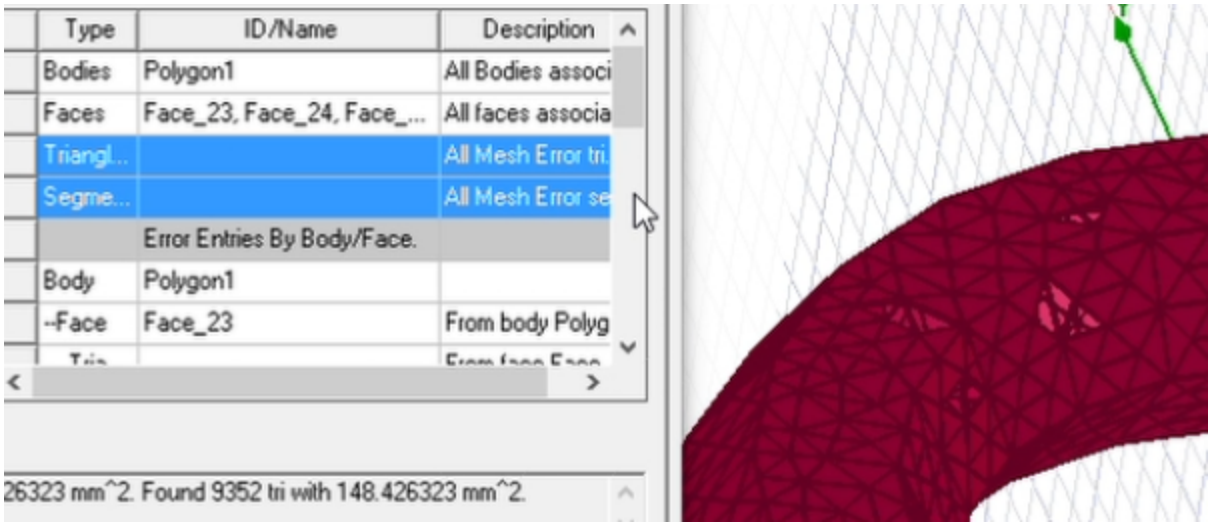
3. On the upper right table, select the body, face, or triangle desired, scrolling the list to find any object, face, triangle, or area of interest.

   Click Face, to see the selected face.

   Select Triangles to see the actual triangles created.

   Select Segment to show the triangles attempted.

   The following figure shows Triangle and Segment both selected (using shift-click). Visually, the unfilled triangles represent places where the triangle is not on the desired layer. However the mesh is still very good and supports an accurate solution.



4. Check boxes allow you to **Hide All Other Objects** checked by default, and **Auto zoom to selection**, unchecked by default.

Related Topics

Plotting the Mesh

Mesh Feedback Tab

Generating the Mesh without Solving

Technical Notes: Skin Depth-Based Mesh Refinement

Technical Notes: Seeding the Mesh

Technical Notes: Guidelines for Seeding the Mesh

Technical Notes: The Mesh Generation Process

Assigning Edge Cut Mesh Operation

1. Select the faces you want Maxwell to refine.

   Alternatively, select an object if you want Maxwell to refine every face on the object.
2. Click **Maxwell 3D or 2D>Mesh>Assign Mesh Operation>Inside Selection>Edge Cut Based**.

   The **Edge Cut Based Refinement** dialog box appears.
3. Type a name for the mesh operation in the **Name** box, or accept the default name.
4. Specify the **Layer Thickness**, and select the units.
5. Click **OK**.

When the mesh is generated, the refinement criteria you specified are used. When the maximum number of elements is reached, some elements may exceed the requested maximum element length.

For 2D designs, edge cut works only if the Clone Mesh works. Refer to *Layer Mesh Generation for 3D Rotational Models* for information on edge cut for 3D designs.

You can also specify Initial Mesh Settings to apply to all objects; however, if you apply separate surface approximation mesh operations to specific objects, the object settings take precedence over the general setting.

Related Topics

Assigning Length-Based Mesh Refinement Inside Objects

Generating the Mesh without Solving

Plotting the Mesh

Technical Notes:Length-Based Mesh Refinement

Technical Notes: The Mesh Generation Process

Technical Notes:Seeding the Mesh

Technical Notes:Guidelines for Seeding the Mesh

Assigning Cylindrical Gap Treatment

1. Select the object you want Maxwell to refine.
2. Click **Maxwell 3D>Mesh>Assign Mesh Operation>Cylindrical Gap Treatment** or **Maxwell 2D>Mesh>Assign Mesh Operation>Cylindrical Gap Treatment**.

   A **CylindricalGap** mesh operation is applied to the selected object, and a **CylindricalGap** icon appears in the Project Manager Mesh folder.

   The CylindricalGap operation only works for the TAU (tolerant) mesher. It is a specific mesh operation assigned to the “Band” object for a rotational transient simulation.

   While you can select any object and assign the CylindricalGap operation to it in any Maxwell simulation, the CylindricalGap operation is effective only for rotational transient simulations for which cases the TAU (tolerant) mesher will be chosen as the default mesher; or for which you have manually selected the TAU(tolerant) mesher.

   The TAU (tolerant) mesher analyzes the “Band” object, calculates the surface normal angle for the band rotational surface, and automatically assigns the desired mesh operation; or adjusts the existing mesh operations to the adjacent true surfaces at both static and moving sides to ensure a smooth and uniform mesh for the “Band” mesh.

   If the CylindricalGap mesh operation is assigned to a regular object, TAU analysis finds the object is not suitable for transient mesh, in which case the regular TAU mesh will be applied. A warning message is then displayed in the message window.

Note	• There can be only one cylindrical gap operation per design. • The operation cannot be assigned to multiple objects.

3. If you wish to change the name of the mesh operation, you can double-click on the **CylindricalGap** icon to open the **Cylindrical Gap Mesh Operation** dialog box in which you can type a name for the mesh operation in the **Name** box.
4. Optionally for 3D transient simulations of rotational models only, if you wish to apply a clone mesh operation, check **Clone Mesh**. This option is unchecked by default.
    - When **Clone Mesh** is selected, the **Band Mapping Angle** setting is enabled. The valid range of angles is 0.1 through 3.0 degrees, inclusive. The default band mapping angle is 3 degrees. (For more information on this setting, refer to *Band Mapping Angle*.)

Note	If the model is not suitable to make a clone mesh or the clone mesh process fails, Maxwell automatically falls back to the regular TAU 3D mesher. Because the clone mesh is generated in the initial mesh, the initial mesh is deleted and regenerated once this option is changed and you click **OK**.

    - When **Clone Mesh** is selected, the **Number of Layers** setting is enabled. You can input the number of mesh layers on the **Moving Side** and **Static Side**, respectively. The valid ranges for these settings are integers ≥ 1. The default values are 1. (For more information on this setting, refer to *Layers in Band Mesh*.)

Related Topics

3D Clone Mesh Generation
Band Mapping Angle
Layers in Band Mesh
Assigning a Band of Motion
Reassigning a Band of Motion

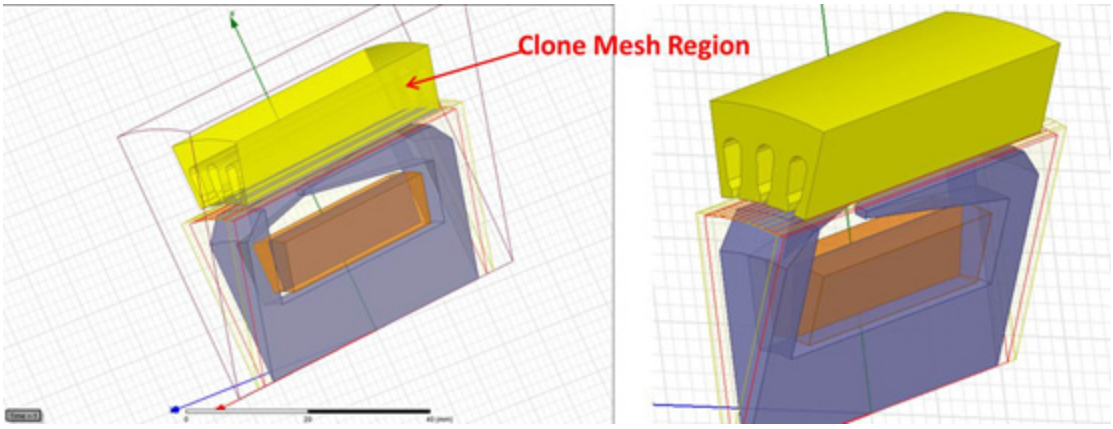
3D Clone Mesh Generation

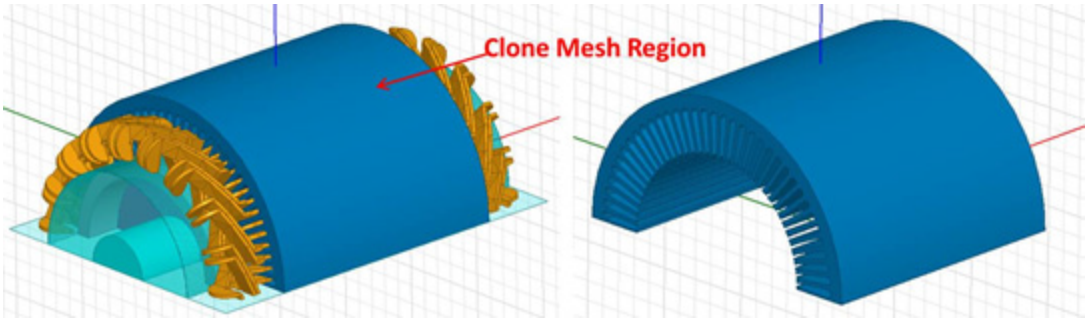
The 3D clone mesh is generated only when the following three conditions are met:

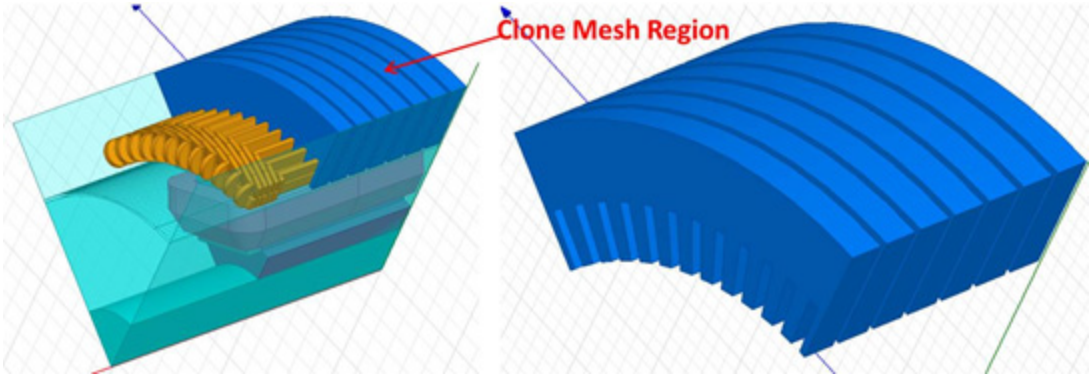
- The Clone Mesh check box is checked.
- The CylindricalGap mesh operation has been assigned.
- The model has a non-skewed stator with more than two teeth.

Note	When Clone Mesh is used, the only additional mesh operation that can be used is surface approximation. If any other mesh operation is used, the clone mesh operation will be ignored.

The solid bodies adjacent to the band are detected to identify the clone object. If the object has more than two teeth that are close to the band, the teeth will be verified to see they have the identical geometric features. If the teeth are skewed, TAU will fall back to the regular mesh. Otherwise the region from the radius of teeth to the outer edge of the object will be further classified and verified to see if the object has the multiple 3D identical sub-regions.





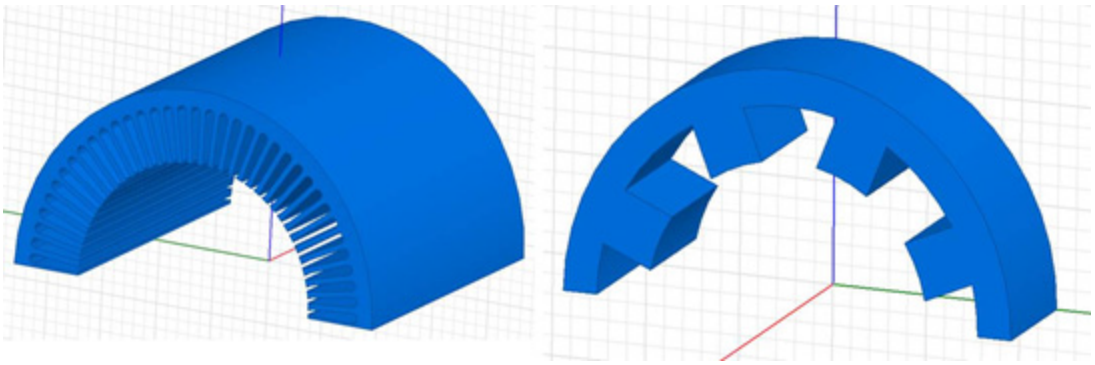


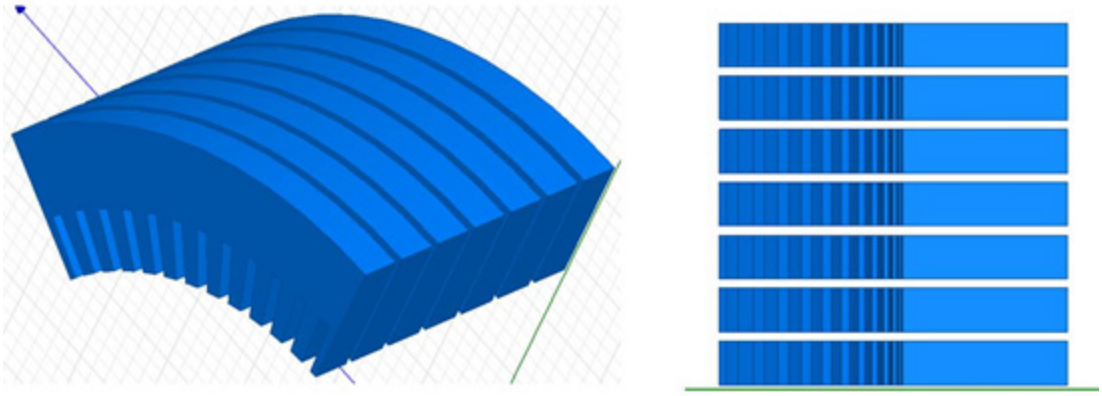
If a 3D clone region has been identified, TAU will classify the 3D rotational model into a static region and a moving region by the band object radius. The Static region includes the clone region and skewed coils regions at one side or two sides. Clone mesh method will then be used to generate the mesh in the clone region, while the TAU regular mesh method will be used to generate the mesh at other skewed regions.

The moving region includes skewed coils regions in which the TAU regular mesh method is used to generate the mesh, and also the non-skewed sweep regions in which the TAU sweep mesh method is used to make the mesh.

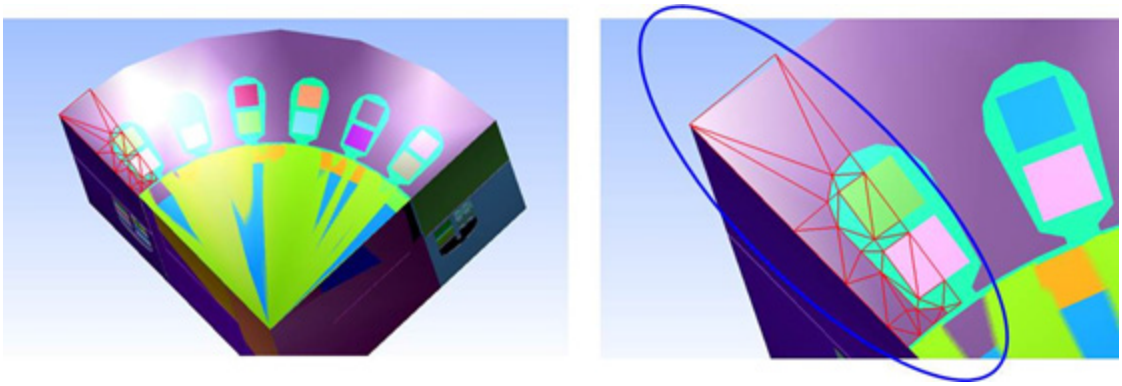
3D Clone Mesh

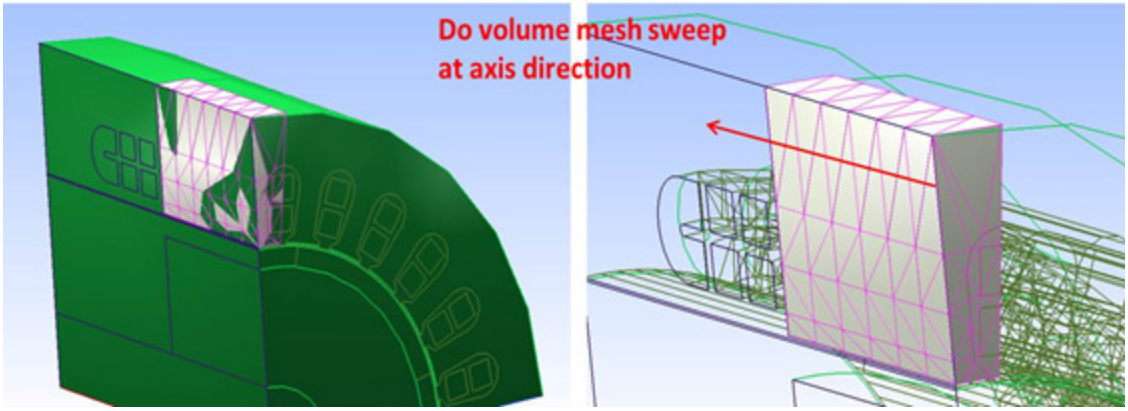
A 3D clone region may have only one simple object, or several sections of clone objects with the gap between as shown in the following images.





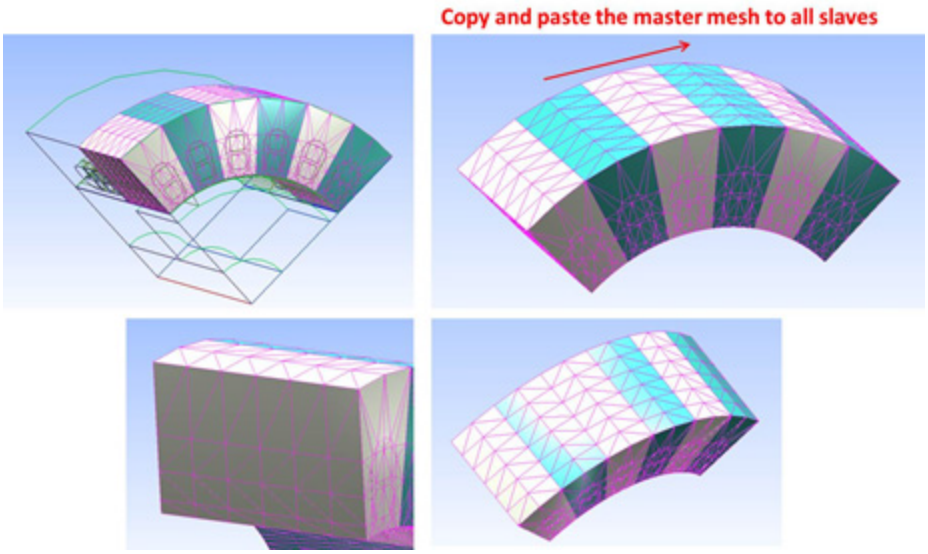
To ensure the identical volume mesh in each sub-region, the volume mesh is generated in the Master region, then by using a copy-and-paste method to duplicate the mesh at all child regions. In the 3D master region, if the region is also symmetric, then a symmetric mesh is generated. “Sweeping mesh” technology is used to make the volume mesh in the master region. In the half of the master region, a surface mesh is generated at the axis side. The volume mesh is generated by sweeping the surface mesh in the axis direction to fulfill the clone master region.





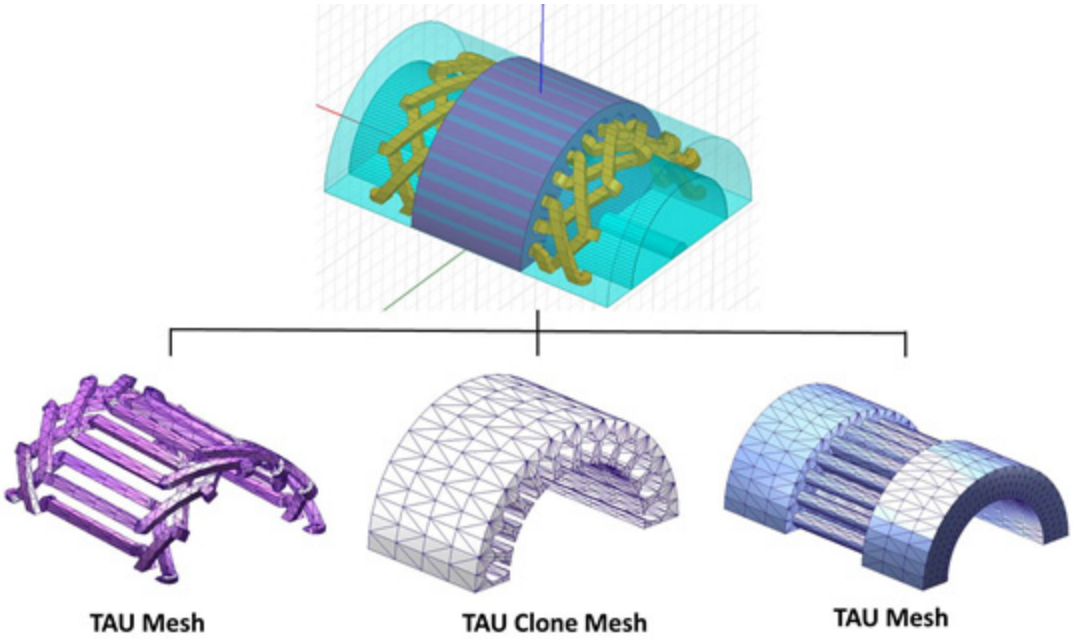
The volume mesh will be mirrored to the other half of the master region to produce the symmetric mesh in the master region.

The volume mesh in the master region is then duplicated and translated to the child regions with copy and paste technology in TAU to complete the 3D clone region mesh.



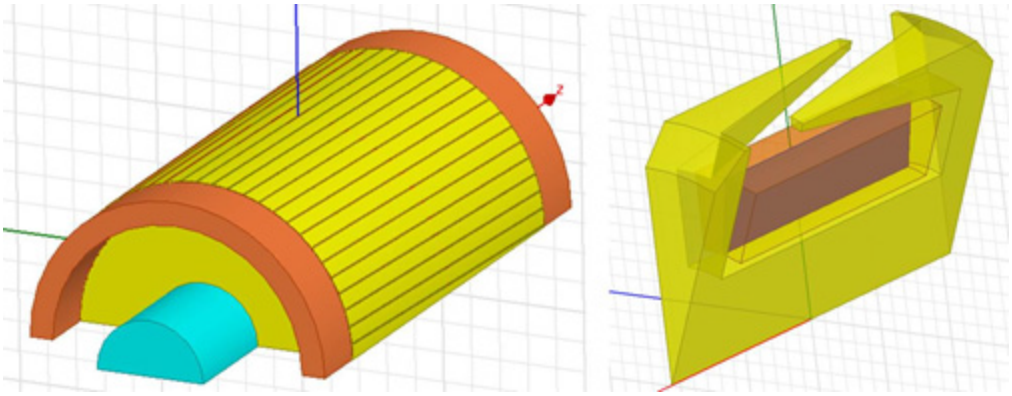
3D TAU Mesh

Beyond the 3D clone region, normally the regions have skewed coils. Therefore TAU regular mesher will be used to make the mesh.

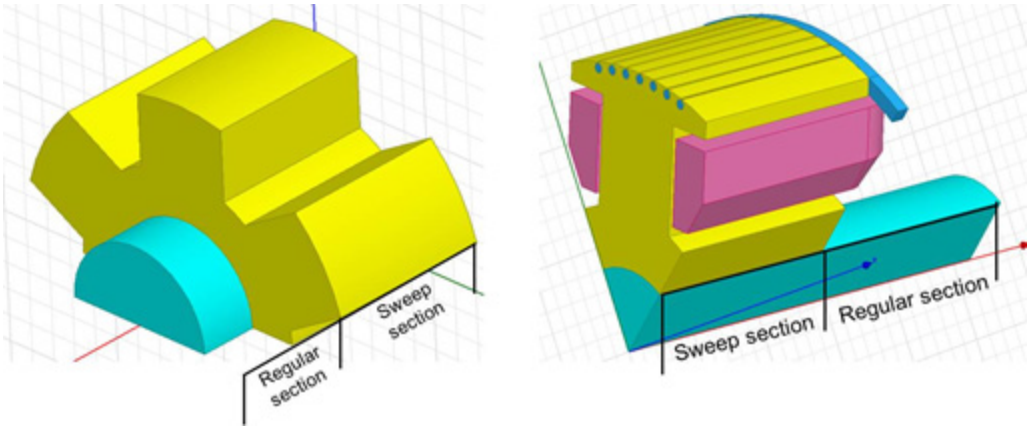


3D Sweep Mesh

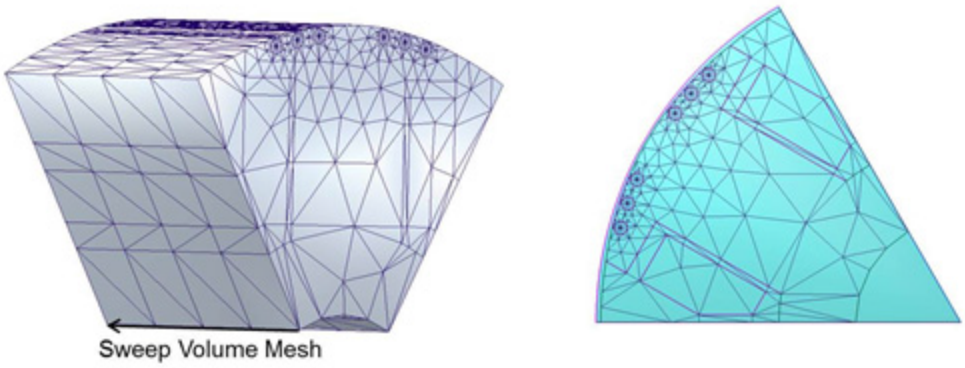
The rotational model for a moving region may have different structures. It may have all skewed coils and rotors. In this case, the TAU regular mesh will be used to make the mesh.



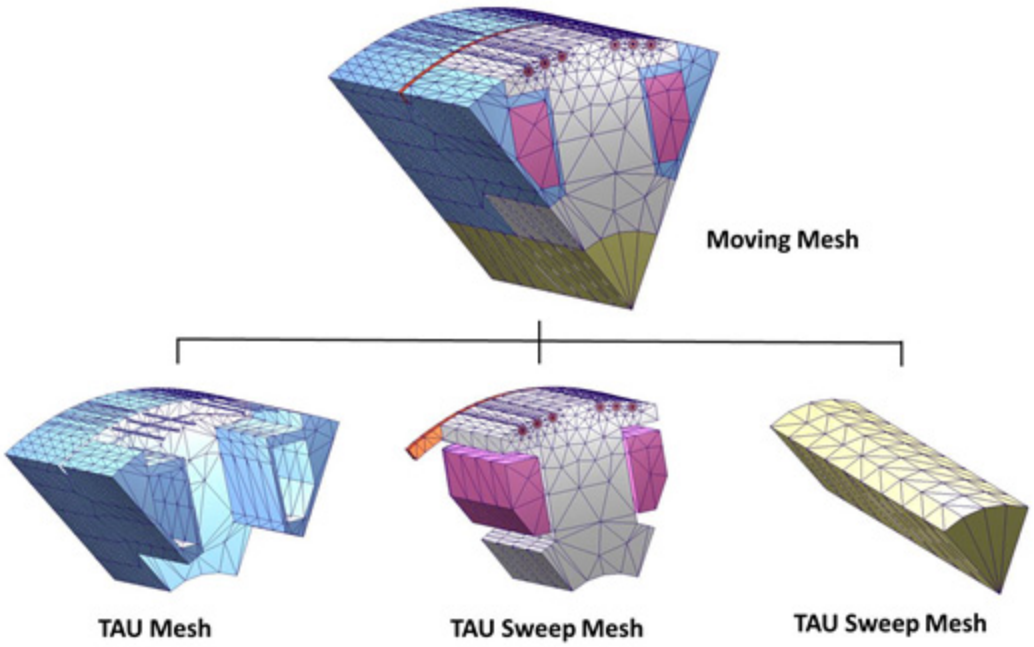
At the axis direction, the moving region can be classified into several specific regions. If the region has skewed objects, it is called a “regular section,” and the TAU regular mesh is applied. If the region can be geometrically sweeping at the axis direction, the region is called a “sweep region,” and the TAU sweep mesh method is applied.



For the “sweep section,” a surface mesh is generated at the side, then the volume mesh is made by sweeping the surface mesh in the axis direction. This ensures a uniform mesh in the rotational direction, while maintaining the long aspect ratio in the axis direction to keep the mesh size small.



The volume mesh for the rest of the regions is made by the TAU regular mesher.



Related Topics

Assigning Cylindrical Gap Treatment

Assigning a Band of Motion

Reassigning a Band of Motion

Band Mapping Angle

Because mesh mapping errors can occur at the interface between the band of motion and the stationary objects when mesh segmentation is performed, the band mapping angle allows these errors to be corrected by making the rotating angle per time step equal to the angle of each mesh segment in the direction of rotation. Thus, the band mesh will match the mesh on the stationary region for each time step. Reducing the mesh mapping error also reduces the noise for the torque calculation and improves simulation accuracy.

The **Band Mapping Angle** setting on the Cylindrical Gap Mesh Operation dialog box, enabled when **Clone Mesh** is selected, supports 3D transient designs for which:

- The band of motion is a true surface.
- The model is not skewed.
- The model is suitable for clone mesh – that is, the model can be segmented into identical sections in the direction of rotation.

The recommended band mapping angle (the angle the rotor rotates in one time step) typically equals the rotational speed multiplied by the time step. For example, if the rotational speed is 1.5 degrees per second and the time step is 2 seconds, then the band mapping angle should be 1.5 x 2 = 3 deg.

The starting position of the simulation should also be considered. In the above example, if the starting position is 0 deg or 3 deg, then 3 deg would be a good band mapping angle. If the starting position is 1.5 deg, the mesh will not be mapped at the starting position, in which case 1.5 deg is the recommended setting.

Another approach for determining the band mapping angle is to start with the model. For example, assume that the stator has 90 slots in the full model, with the solution domain angle equaling 1/6 of the full 360 degree model or 15 slots for 60 mechanical degrees. If you would like to solve with each slot having 8 mesh segments, there would be 120 total mesh segments (15 slots x 8 segments per slot). In this case, the band mapping angle would be 60 (deg) /120 = 0.5 deg. To ensure rotation through one mesh segment per time step, assuming the rotational speed is 1200 rpm, the time step should be set as:

$$Time\ step = \frac{1}{(speed\_rpm/60_{seconds}) \times total\_segment\_number \times (360^{\circ}/domain\_angle^{\circ})}$$

$$Time\ step = \frac{1}{(1200/60) \times 120 \times (360/60)} = \frac{1}{14400}\ seconds$$

Related Topics

Assigning Cylindrical Gap Treatment

Assigning a Band of Motion

Reassigning a Band of Motion

[BETA] Maxwell 2D Band Mapping Angle

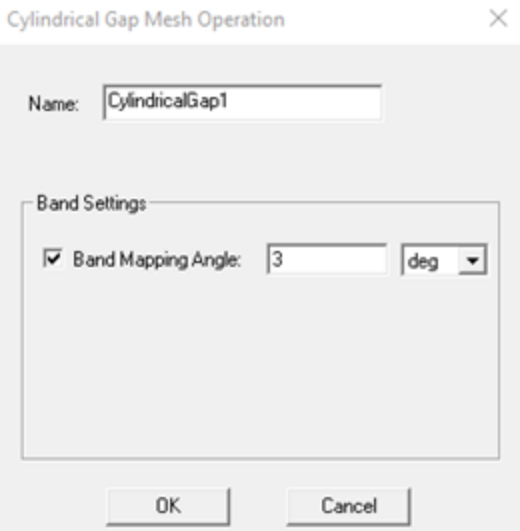
Band Mapping Angle is a new meshing option for transient simulation of motors in Maxwell 2D. When selected, the meshing process will ensure that each mesh segment on the outside of the band (the curve that separates the static and moving regions) has exactly the sweep angle that has been specified by the user. For example, for a quarter model with 90 degrees of matching boundaries, specifying a band mapping angle of 0.5 degree will result in a band outline having 180 mesh segments of equal length.

Because the meshing angle is determined by this setting, the band object must have a curved outline, using zero segments.

To enable this feature, select **Maxwell 2D Band Mapping Angle** from the **Beta Options** dialog, which can be accessed in the Desktop options. Enabling this feature will cause Electronics Desktop to automatically restart.

To use 2D band mapping angle:

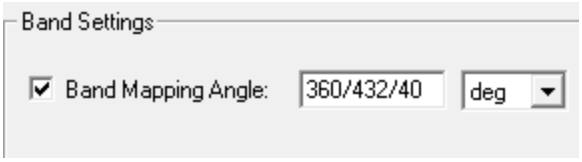
1. When you open a 2D transient project, go to **Project Manager> Mesh>CylindricalGap**, then right-click **Property**; the Cylindrical Gap Mesh Operation dialog box will be shown as:



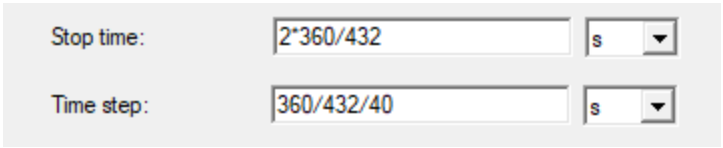
2. Select **Band Mapping Angle**, and set the desired segment angle for the band mesh, then click **OK**.

   > **Tip:** It is best to synchronize the transient simulation time step with the band mapping angle.

   For example, suppose the following mapping angle is used to accurately model the cogging torque:



If the motor's rotational velocity is 1 degree/second, then the following time step should be used:



This practice will help to remove numerical noise in the results.

There are two ways to verify that the band mapping angle has been applied to a simulation:

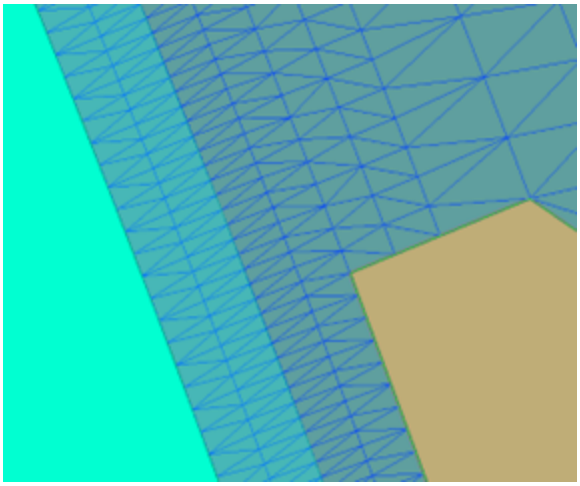
- The specified angle will appear under Initial Meshing, on the Mesh line after "TAU":

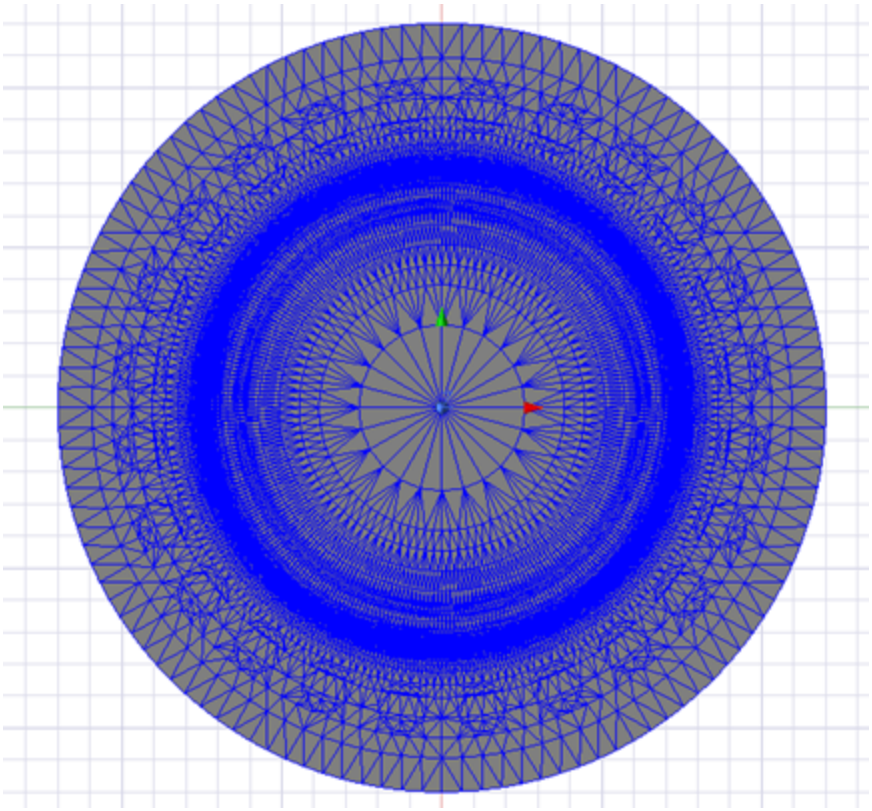
- The effects of the mapping angle will be seen in mesh plots:

  A successful band mapping angle mesh has several features in the mesh:

  - An additional layer will be created on both sides of band.
  - The triangle orientation will be exactly the same on each side of band, to eliminate noise.
  - The layered structure will extend to the moving and static regions.

- Mesh density will smoothly transition from fine to coarse with distance from the band.





Notes

- This feature is intended to replace the use of wire bodies to guide meshing in the band. However, if wire bodies are included, the mesh will respect them along with the band mapping angle.

- Mesh density control is planned for a future release. In the meantime, seed refinement or surface approximation settings can be used to control the mesh density in certain regions.
- Clone meshing is not supported. There may be some asymmetry in the meshing of teeth or slots.
- Eccentricity is not supported.
- This feature is limited to Maxwell 2D transient models with a Cylindrical Gap mesh operation.

Related Topics

Assigning Cylindrical Gap Treatment

Layers in Band Mesh *(Maxwell 3D Magnetostatic and Transient)*

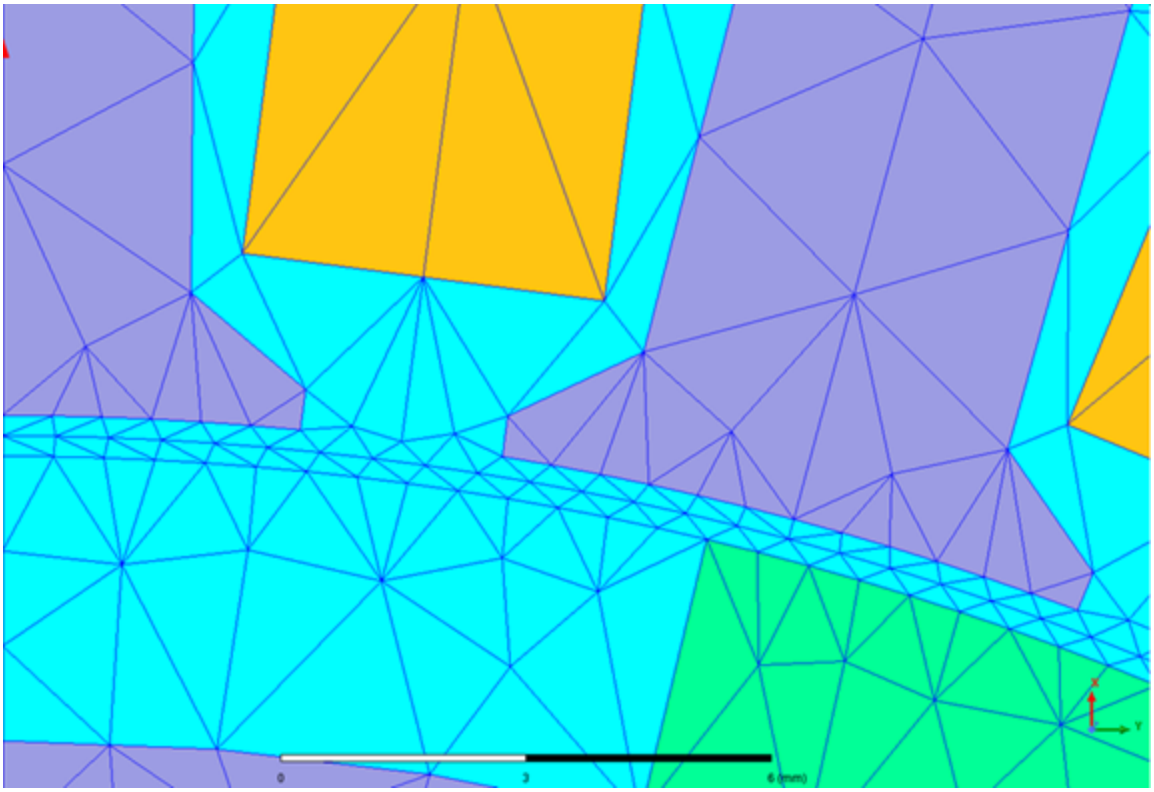
Maxwell rotating machine models need very precise cloned mesh to get accurate estimates of cogging torque. Thus, the mesh near the interface of the static side and moving side, i.e., band surface, is crucial to the accuracy of this simulation result.

Clone mesh is an initial mesh process and currently provides one layer mesh between band and stator and one layer mesh between band and rotor. However, one layer may not provide sufficient accuracy for some designs. For such cases, multiple layers in band for clone mesh can be used to improve accuracy. By assigning this mesh operation, you can control the number of layers near the band on *both* the static and the moving side without losing the repeatability of the clone mesh.

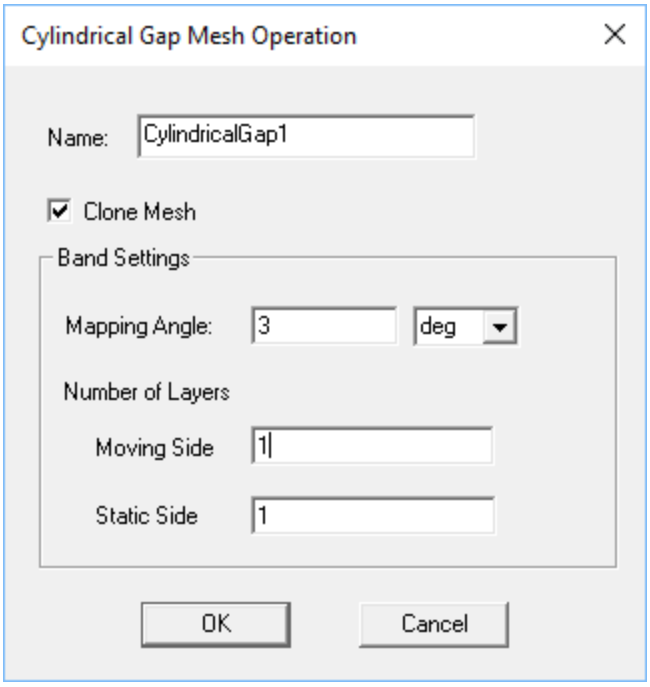
Note	This feature is applicable only to Maxwell 3D magnetostatic and transient solution types.

Example

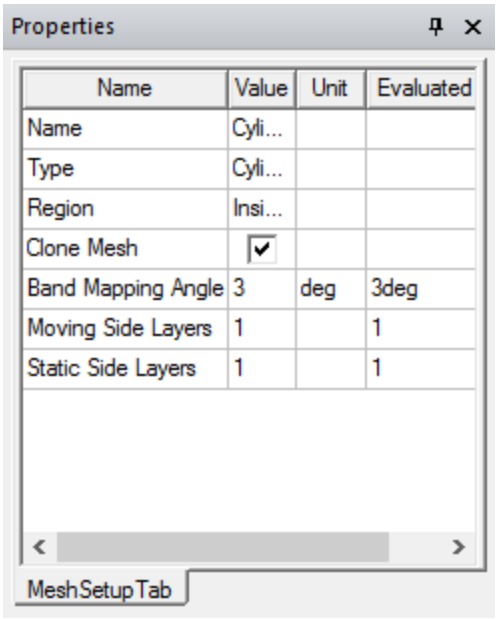
Without multiple layers in the band mesh, when the model is cloned, the static side has clone geometry for which clone mesh is provided. The moving side does not have clone geometry so sweeping mesh is provided for it. This simple mesh type is shown below.



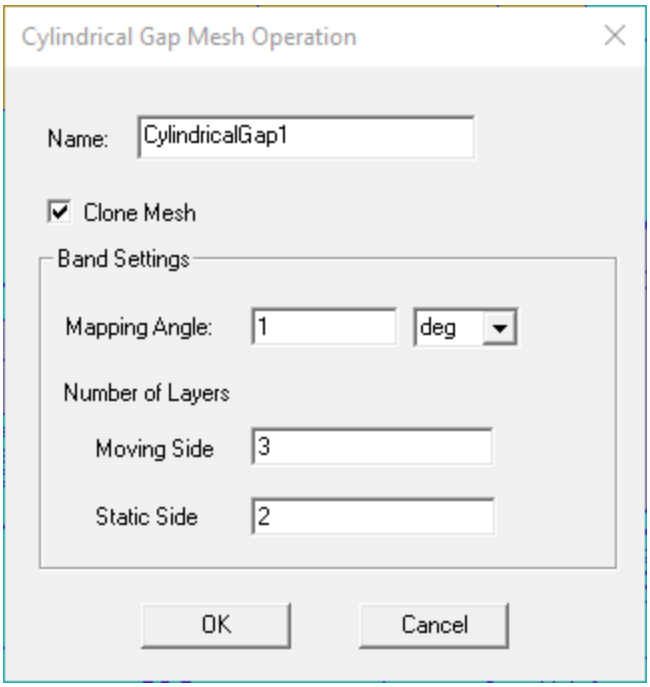
To improve accuracy of results, multiple layers can be specified for a **Clone Mesh** in the **Band Settings** section of the **Cylindrical Gap Mesh Operation** dialog box.

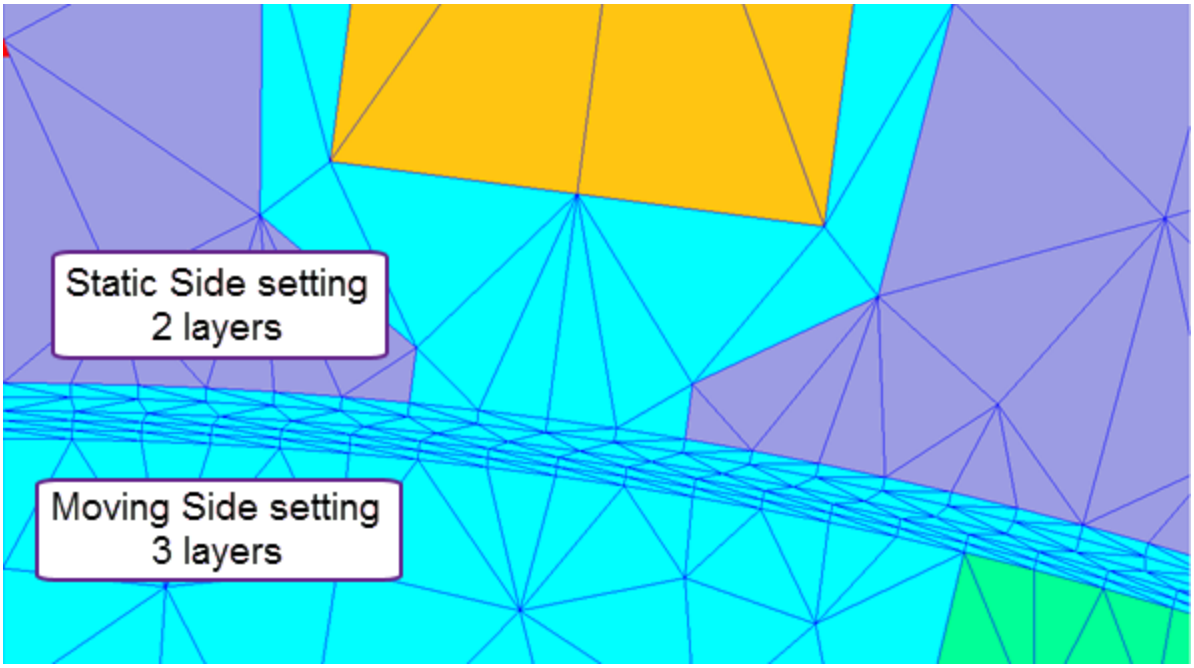


The layers options are also available in the **Properties** window **Mesh Setup Tab** when the cylindrical gap mesh operation is selected in the project tree.



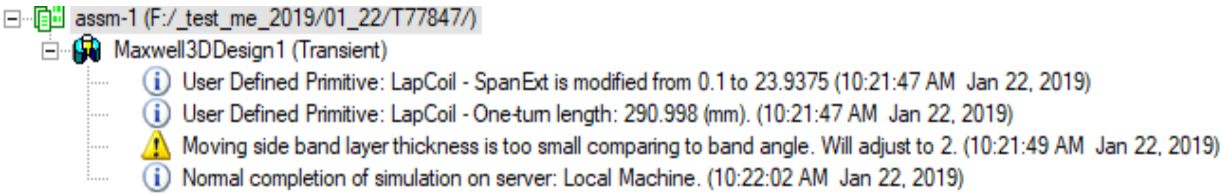
By setting the **Number of Layers** for the **Moving Side** and/or the **Static Side** to integer values greater than 1, the number of band layer meshes will be increased accordingly.





Considerations and Limitations

- If you specified a number of layers that would result in a lowering of mesh quality, clone mesh will adjust the number of layers, and give a warning message similar to the following:



- If the model it not suitable for clone mesh, or if clone mesh fails to mesh it, this setting will not be respected in TAU regular 3D mesh. A warning message will be given by mesh in this situation.
- Post processing seeding refinement will respect the layers in the mesh. Adding any post processing mesh operation will not destroy the layers; however, the clone mesh might be destroyed.

Related Topics

Assigning Cylindrical Gap Treatment

3D Clone Mesh Generation

Assigning a Band of Motion

Reassigning a Band of Motion

Clone Mesh Density

For Maxwell 3D rotational transient design, this meshing operation allows you to specify the clone mesh density while keeping the identical mesh in identical sub-regions. Different from Length-Based Mesh Refinement, this Clone Mesh Density mesh operation will only be applied to clone mesh. Only when clone mesh is enabled under cylindrical mesh operation, mesh will respect this mesh operation. Mesh density can be controlled while identical mesh is maintained.

To control the clone mesh density in Maxwell:

1. Select the object.
2. Click **Maxwell>Mesh>Assign Mesh Operation>Inside Selection>Clone Mesh Density**.

   The **Clone Mesh Density** dialog box appears.
3. Type a name for the mesh operation in the **Name** box, or accept the default name.
4. To specify the radial direction density, enable the radial direction option. Type the maximum length of the element size required inside the object(s) in the **Maximum element length** box, and select the units. Maxwell refines the element edges inside the object(s) until they are equal to or less than this value. The default value is set to 10% of the maximum dimension of the bounding box of the model.
5. To specify the axial direction density, enable the axial direction option. Enter the **Number of layers** to be added. Maxwell refines the mesh in axial direction inside the object(s) until the number of mesh layers in the axial direction are equal to or more than this value.
6. Click **OK**. You need to specify mesh density in at least one direction to enable this mesh operation.

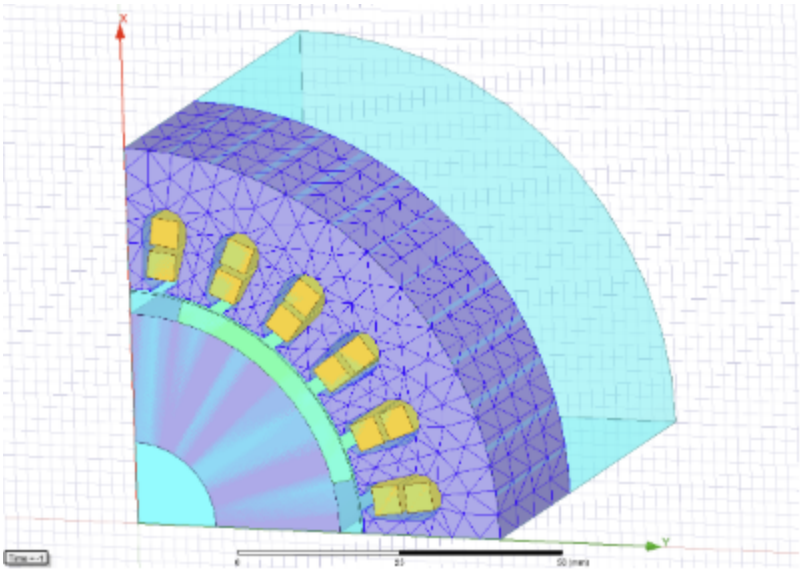
When the clone mesh is generated, the refinement criteria you specified are used. You can specify different maximum length of the element size on different objects. Note that identical density setting should be applied to identical objects to ensure the specified density to be applied on those objects.

You can also specify different axial direction number of layers on different objects. Note that the number of layers in axial direction are maintained separately inside and outside of the band. If objects share a same region in axial direction on the same side of the band, the mesh refinement will refines until the layer thickness equal or smaller than the smallest average layer thickness in this region.

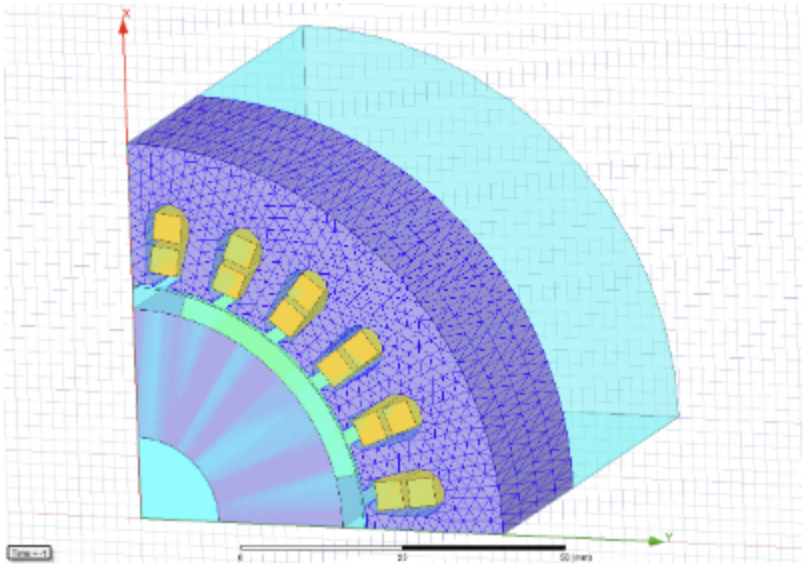
The Clone Mesh Density can not be applied to band object. To control the element size on band, use the **Band Mapping Angle** setting.

Clone Mesh Density can only be respected in clone mesh region. Mesh density in regions that are skewed or not clonable may not be controlled as precise as it is in clone region.

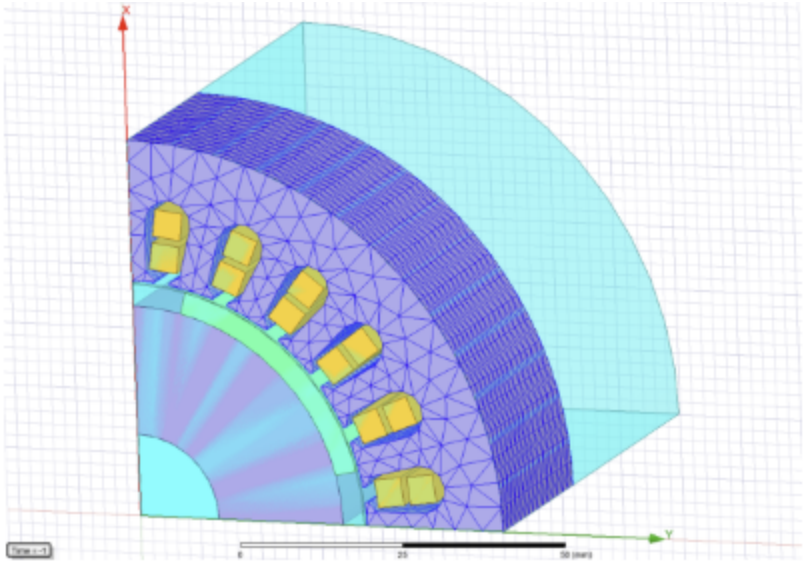
Original clone mesh without Clone Mesh Density mesh operation:



Clone mesh with Clone Mesh Density mesh operation on Stator. Radial direction mesh is refined according to maximum element length specification:



Clone mesh with Clone Mesh Density mesh operation on Stator . Axial direction mesh is refined according to number of layers specification:



Related Topics

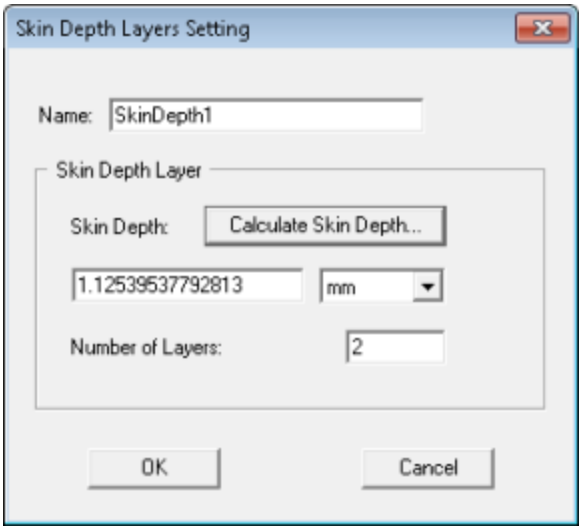
Assigning Length-Based Mesh Refinement on Object Faces
Generating the Mesh without Solving
Plotting the Mesh
Technical Notes:Length-Based Mesh Refinement
Technical Notes: The Mesh Generation Process
Technical Notes:Seeding the Mesh
Technical Notes:Guidelines for Seeding the Mesh

Skin Depth Layers Setting for 2D

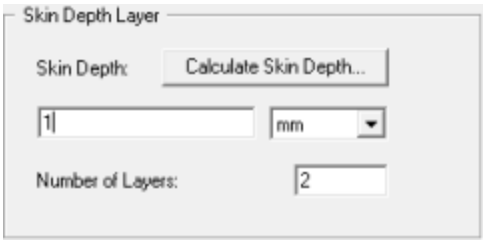
You can apply these settings only to the edges of sheet bodies in Maxwell 2D when the TAU mesher has been selected in the **Initial Mesh Settings** dialog.

Note	Any changes you make using these settings will invalidate the initial mesh, as well as the current mesh and solutions, if any.

1. Select the edge of a sheet body, and click **Maxwell2D>Mesh>Assign Mesh Operation>On Selection>Skin Depth Based...** to open the **Skin Depth Layers Setting** dialog:

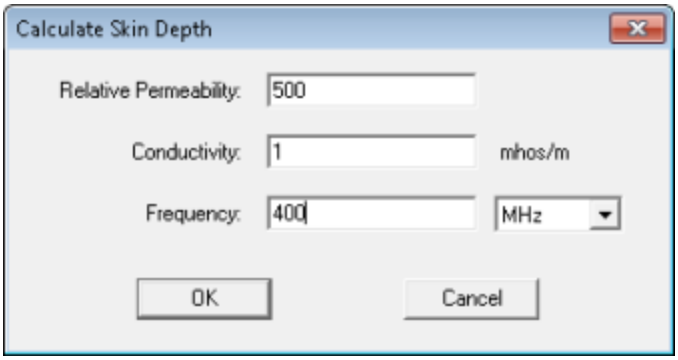


2. Type a name for the mesh operation in the **Name** text box or accept the default name.
3. Type the skin depth within which to refine the mesh in the **Skin Depth** text box.



Alternatively, calculate the skin depth based on the object's material permeability and conductivity and the frequency at which the mesh will be refined:

a. Click **Calculate Skin Depth**.
   The **Calculate Skin Depth** dialog box appears.



b. Enter the material's **Relative Permeability** and **Conductivity**.
c. Specify the **Frequency** at which to refine the mesh.

d. Click **OK**.
Maxwell calculates the skin depth and enters its value in the **Skin Depth** text box.

4. In the **Number of Layers of Elements** text box, type the number of layers to add perpendicular to the object's surface.

Maxwell will add an equivalent number of mesh points to each layer. For example, if Maxwell added 10 points to satisfy the **Surface Triangle Length**, it will add 10 points to each layer.

Note	Caution should be used with this mesh operation, as very thin layers may cause a reduction in mesh quality or unnecessarily cause the generation of a very large mesh. Furthermore, regions refined under this operation and its close neighbors do not participate in solution adaptive refinement. This is another reason to use this seeding operation with caution.

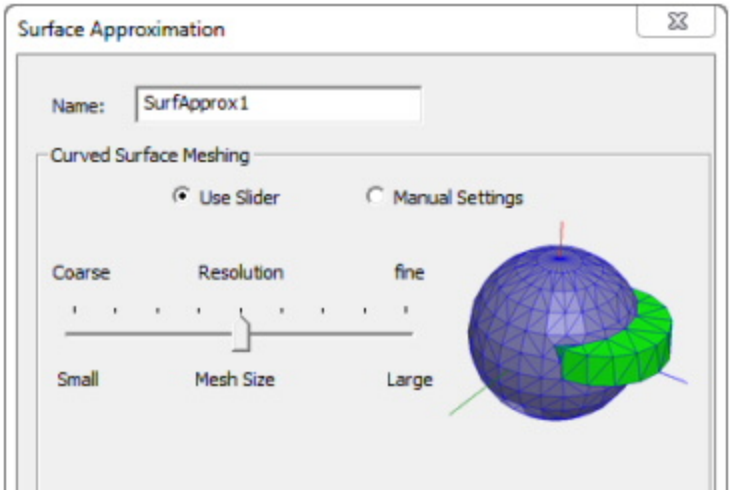
Related Topics

Defining Mesh Operations
Specifying Initial Mesh Settings
TAU 2D Skin Layer Mesh Generation

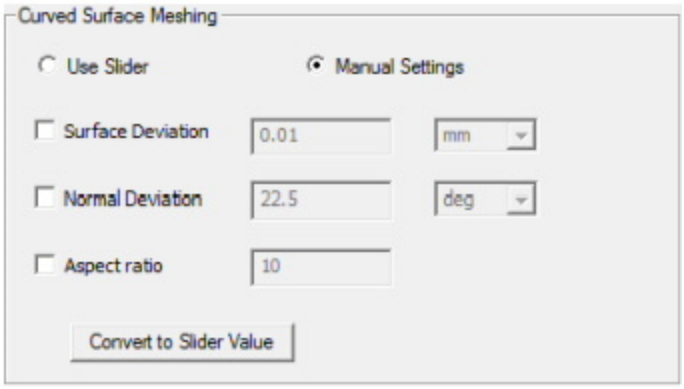
Modifying Surface Approximation Settings

Maxwell applies surface approximation settings when it generates the initial mesh. If you modify Maxwell's default settings after the initial mesh has been generated, they do not affect the mesh for that design variation. Surface approximation is useful, for example, for curved surfaces, for which the mesh will not exactly reproduce the surface shape. It also can be used to restrict the aspect ratio of triangles on planar surfaces.

1. Select the faces for which you want to modify the surface approximation settings.
   - Alternatively, select an object if you want to modify the surface approximation settings of every face on the object.
2. Click **Maxwell>Mesh>Assign Mesh Operation>Surface Approximation**. to open the **Surface Approximation** dialog box.



3. Type a name for the group of settings in the **Name** box, or accept the default name.
4. Under **Curved Surface Meshing**, you can select **Use Slider** or specify **Manual Settings**. The slider includes a visual representation of the resolution you choose, ranging from Coarse Resolution with a Small Mesh Size through a nine position scale to a fine resolution with a Large Mesh Size.
5. If you choose Manual Settings, the dialog changes to show text fields. (If you select **Use dynamic surface resolution** in the Initial Mesh Settings, use of **Manual Settings** is not permitted. You can only use slider bar to specify three levels of surface representation as: coarse (1-3), normal (4-6), and fine (7-9).)



Note	If you select **Use dynamic surface resolution (Beta)** in the Initial Mesh Settings, use of **Manual Settings** is not permitted. You can only use slider bar to specify three levels of surface representation as: coarse (1-3), normal (4-6), and fine (7-9).

6. If you select **Surface Deviation (length)**, type the distance between the true surfaces of the selected faces and the meshed faces in the text box, and select the units.

7. If you select **Normal Deviation (angle)**, type the angular distance between the normal of the true surface and the corresponding mesh surface, and select the units.
8. If you select **Aspect ratio**, type a value in the box. This value determines the shape of the triangles. The higher the value, the thinner the triangles. Values close to 1 result in well-formed, wide triangles.
9. Click **OK**.

   The settings are applied to the initial mesh generated on the selected surface. The group of settings is listed in the project tree under **Mesh**.

You can also specify Initial Mesh Settings to apply to all objects; however, if you apply separate surface approximation mesh operations to specific objects, the object settings take precedence over the general setting.

Note: Checking **Use dynamic surface resolution (Beta)** in the Initial Mesh Settings specifies the best-practice mesh operations over the geometric models. The default mesh operations or user defined mesh operations may or may not be replaced by the optimized mesh operations with model analysis. In general, large curve faces, curve faces with small gaps, or skewed cables get more smooth curvature representation while small curve faces, such as fillets, small curve objects, and so forth, get relatively coarse triangulations so that overall mesh size will be reduced. This dynamic surface resolution mesh operation supports 3D volume mesh and surface mesh in all products.

Related Topics

Plotting the Mesh

Technical Notes: Surface Approximation Settings

Technical Notes: Guidelines for Modifying Surface Approximation Settings

Technical Notes: The Mesh Generation Process

Surface Approximation

Object surfaces in Maxwell may be planar, cylindrical or conical, toroidal, spherical, or splines. The original model surfaces are called true surfaces. To create a finite element mesh, Maxwell first divides all true surfaces into triangles. These triangulated surfaces are called faceted surfaces because a series of straight line segments represents each curved or planar surface.

For planar surfaces, the triangles lie exactly on the model faces; there is no difference in the location or the normal of the true surface and the meshed surface. When an object's surface is non-planar, the faceted triangle faces lie a small distance from the object's true surface. This distance is called the surface deviation, and it is measured in the model's units. The surface deviation is greater near the triangle centers and less near the triangle vertices.

The normal of a curved surface is different depending on its location, but it is constant for each triangle. (In this context, "normal" is defined as a line perpendicular to the surface.) The angular difference between the normal of the curved surface and the corresponding mesh surface is called the normal deviation and is measured in degrees.

The aspect ratio of triangles used in planar surfaces is based on the ratio of circumscribed radius to the in-radius of the triangle. It is unity for an equilateral triangle and approaches infinity as the triangle becomes thinner.

You can modify the surface deviation, the maximum permitted normal deviation, and the maximum aspect ratio of triangles settings on one or more faces at a time in the **Surface Approximation** dialog box. (Click **Maxwell>Mesh>Assign Mesh Operation>Surface Approximation**.)

The surface approximation settings are applied to the initial mesh.

Note	For the initial mesh, all the vertices of the triangles lie on the true surfaces. During mesh refinement, all the points are first added to the faceted surface for all products, not to the true surfaces. Then, for all Maxwell projects, and for curvilinear meshing enabled HFSS and HFIE projects, the points on the faceted surfaces will be moved to the curved surface locations when such movement does not degrade the quality of the mesh.

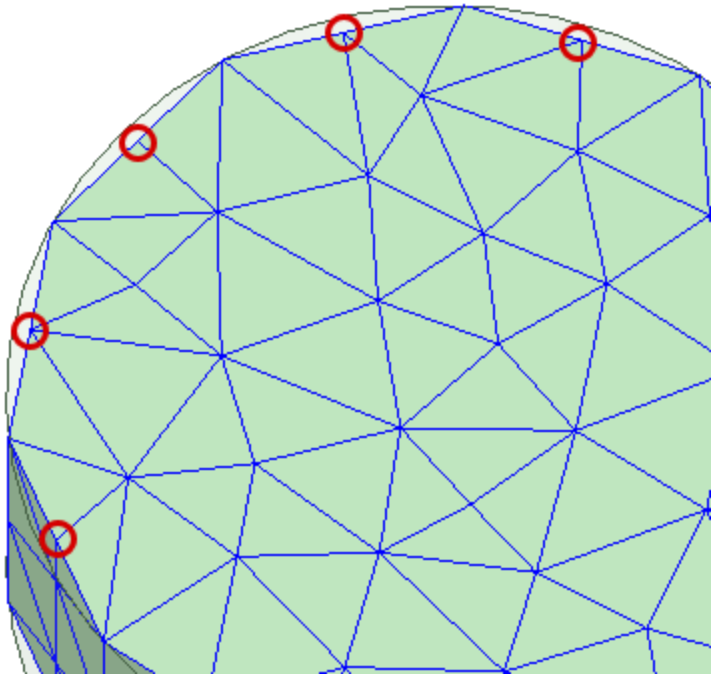
Apply Curvilinear Meshing Command

For **HFSS**, **Maxwell**, and **Mechanical** designs, Curvilinear elements provide an accurate representation of the CAD model geometry by improving the conformity of element nodes to curved CAD surfaces. You can apply curvilinear elements as a Mesh Operation on selected curved model faces see procedure below or as a global option in the Initial Mesh Settings dialog box.

When curvilinear elements are applied globally or to specific CAD faces, the meshing process attempts to pull corner nodes of elements added during mesh refinement to the curved CAD faces. The following figures explain the effects with and without curvilinear elements applied:

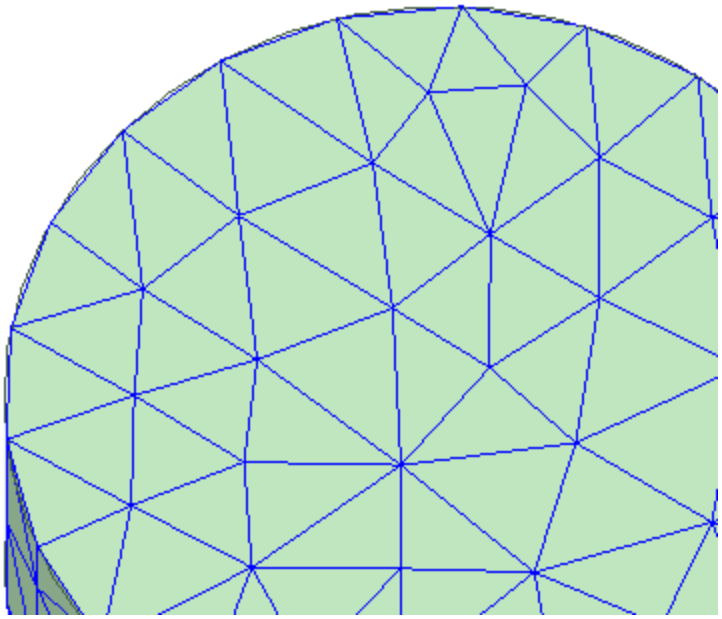
- With the **Apply Curvilinear Elements** option *not* selected:

  Corner nodes of an initial coarse mesh are located on curved CAD surfaces. However, as the mesh is refined, the corner nodes of any additional elements remain along the plane of the intial mesh's element faces. Therefore, the number of facets along a curved face (in the circumferencial direction) does not increase, and the facets remain coarse. This effect is shown in the first mesh plot below. Note the red-circled nodes. These nodes were added as a result of mesh refinement, and they were not pulled to the curved CAD face:



- With the **Apply Curvilinear Elements** option *selected*:

  The meshing process will attempt to pull all corner nodes of additional elements added during mesh refinement to the curved CAD faces. Therefore, the facets along curved faces are finer, and the conformity of the mesh to the curved CAD faces is improved. The effect is as shown in the following, second mesh plot:

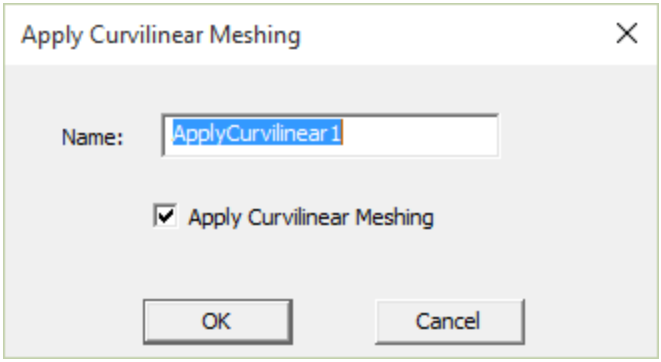


How to Apply Curvilinear Elements to Specific Curved Faces:

You can apply curvilinear meshing as a Mesh operation.

1. Select the object or face to which you want to apply curvilinear meshing.
2. Select **Mesh** in the Project Tree, then right-click for the short cut menu and select **Assign Mesh Operation>Apply Curvilinear Meshing**.

   This displays the **Apply Curvilinear Meshing** dialog.



3. You can accept the default name or provide one of your own.
4. Check the box and click **OK** to apply the curvilinear meshing operation.

Related Topics

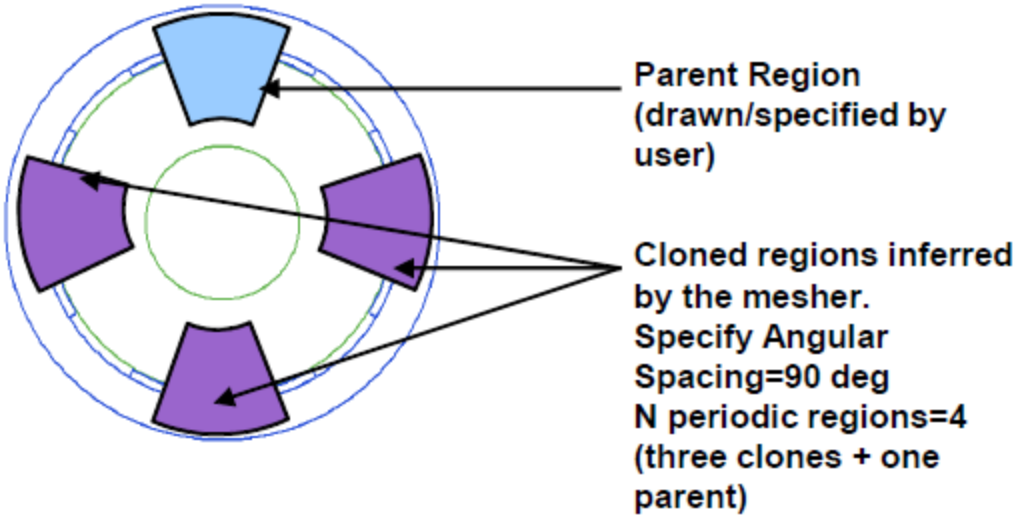
Defining Mesh Operations

Cloning a Mesh

(Cloning a Mesh applies to Maxwell2D Designs only)

The clone mesh operation allows users to use the mesh data from a user-specified parent region of a Maxwell 2D design to apply identical meshes to its cloned regions. In the example shown below, three cloned regions have been generated with an angular spacing of 90 degrees from the user-drawn parent region.

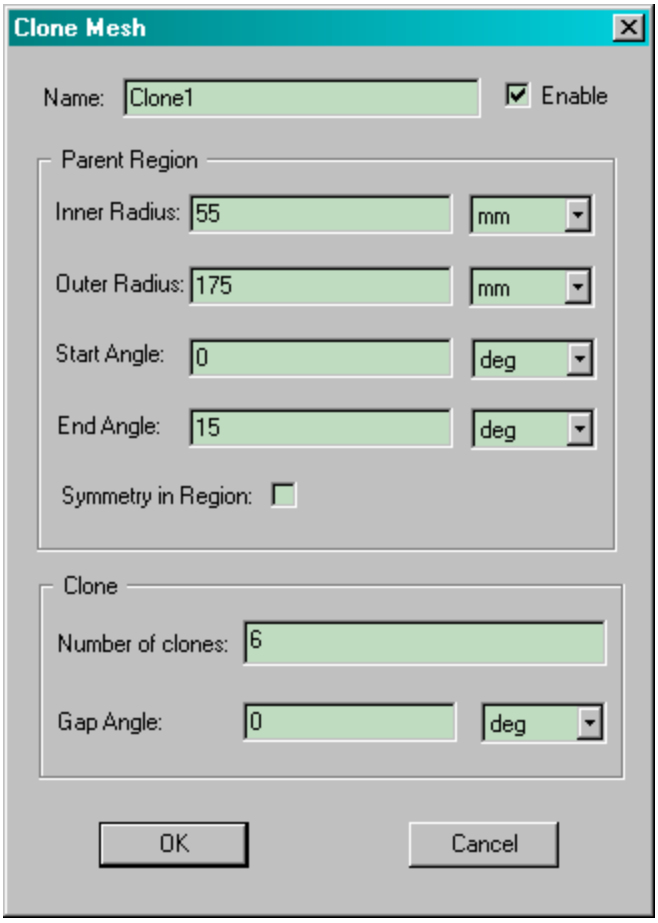
Note	For non-Transient solution types, this operation is applied only on the initial mesh. If you do adaptive refinement, the meshes will no longer match; the operation is not reapplied during adaptive refinement.



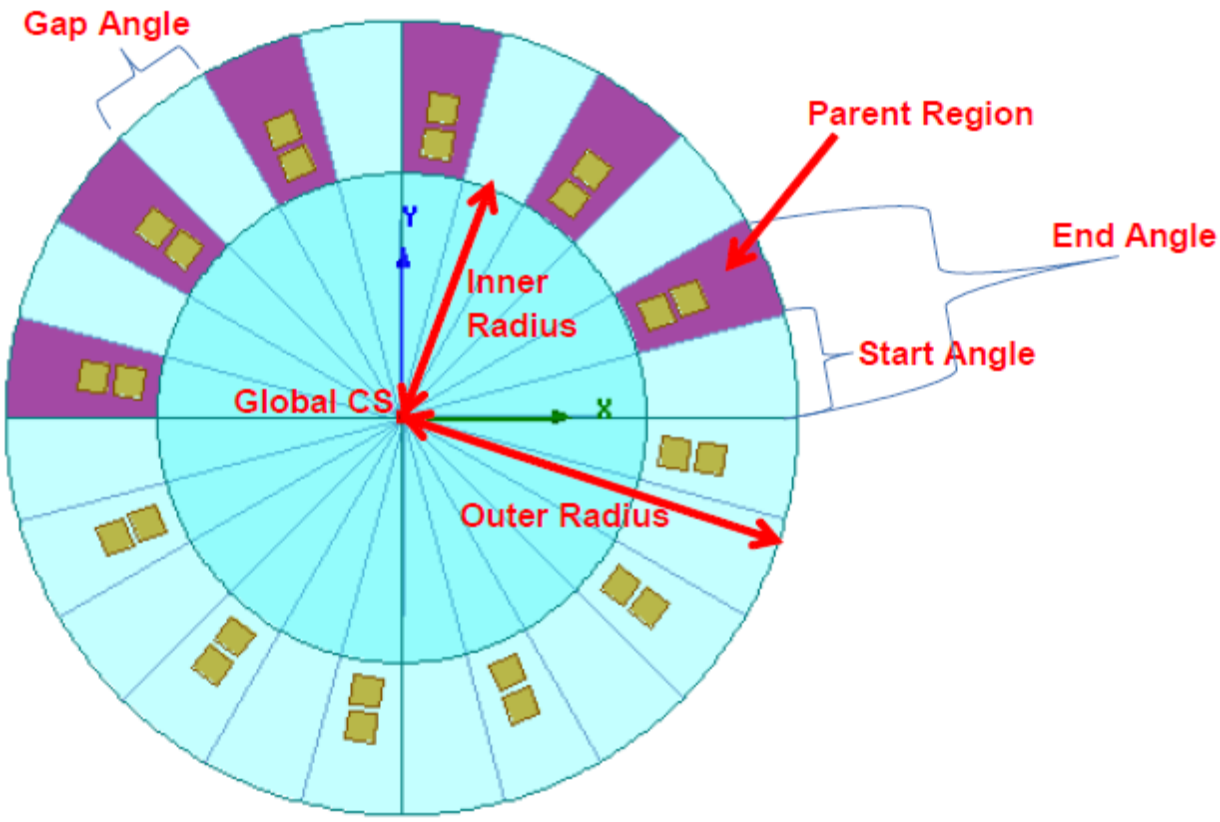
Clone mesh operations are performed after all other mesh operations are done.

Note	To use clone mesh, you must also setup Length-Based Mesh Refinement Inside Objects.

1. To begin a clone mesh operation, select **Maxwell 2D>Mesh>Assign Mesh Operation>Clone Mesh...** to open the **Clone Mesh** dialog box.



2. First set up a **Parent Region**. (Refer to the following figure.)



To set up the parent region, define the following parameters:

- **Inner Radius**
- **Outer Radius**
- **Start Angle** (default=0)
- **End Angle**
- **Symmetry in Region** - treat the region (parent and clone) as symmetric inside the region

3. Specify how to clone the parent region by defining the following parameters:
   - **Gap Angle** (default=0) - angular spacing between clones
   - **Number of clones** - number of periodic regions (including the parent)
4. Like other mesh operations, the user can enter a **Name**, which can be enabled and disabled. Multiple clone mesh operations (with different names) can be created.
5. When finished, click **OK**.
6. To edit the clone mesh setup data, use the dialog or the Properties window.

Modifying the Model Resolution

(Setting Model Resolution applies to Maxwell3D Designs only)

The model resolution parameter is used by the meshmaker to distinguish large features from small features in the model. This setting controls how large a feature must be to be resolved by the meshmaker. For example, if you set the model resolution length to 20 mm, any model features smaller than 20 mm are not represented in the mesh. Neither the model nor the model files are changed. The resolution only controls how the mesh for the model is represented.

While removing small details, if the meshmaker finds that the representation of the model is not accurate enough, it returns an error condition. The meshmaker then starts with the most accurate representation and prunes away the details smaller than the model resolution length. It returns an error if the specified model resolution length forces the final representation to deviate too greatly from the model.

You can set Model Resolution on one or more objects to remove unnecessary details from the mesh representation. This can be used to reduce the mesh complexity of the selected objects.

A cautionary note is needed concerning the use of model resolution. It can sometimes make model faces fail to be represented in the mesh, which can effectively remove some of your boundary conditions. This can only happen if faces are closer together than the model resolution distance you specify.

To modify the model resolution:

1. Select the object or objects on which to specify a Model Resolution operation.
2. Click **Maxwell 3D>Mesh>Assign Mesh Operation>Model Resolution**.

   The **Model Resolution Mesh Operation** dialog box appears.

   Alternatively, you can display the same dialog if you:

   a. Right-click on **Mesh** in the Project Tree to display the shortcut menu.
   b. Click on **Assign Mesh Operation>Model Resolution** on the shortcut menu.

   The **Model Resolution Mesh Operation** dialog contains text fields for the mesh operation Name and radio buttons with choices for the following

   - **Auto Simplify Using Effective Thickness**

     The mesher calculates the resolution length based on each object's effective thickness. One mesh operation can be assigned to many objects, and each will be simplified based on its own dimensions. Use the **Auto Simplify** selection:

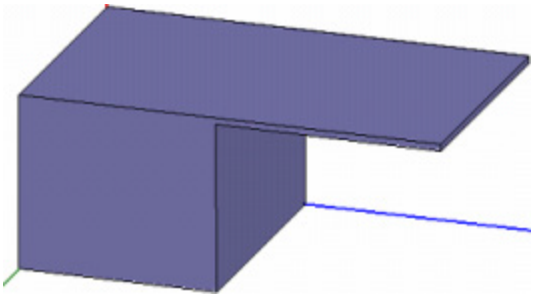
     - To remove many details while retaining an object's overall shape and size
     - For objects of generally uniform thickness
     - To assign one mesh operation to many objects

   - **Use Model Resolution length**

     This enables fields for you to specify the resolution value and units. Use this selection for:

     - Tighter control of mesh accuracy.
     - Objects of non-uniform thickness. For example, the thin section of the object shown

below might be lost with **Auto Simplify**.



3. After defining the operation, click **OK**.

   This adds the named Model Resolution operation under the **Mesh** icon in the Project Tree.

Note	Setting Model Resolution will invalidate any existing solutions. When two objects in contact have different model resolution lengths, the smaller length will apply for the common regions.

Related Topics

Plotting the Mesh

Setting the Healing Options

Technical Notes: *Model Resolution*

Reverting to the Initial Mesh

The initial mesh is the mesh that is generated the first time a design variation is solved. It includes surface approximation settings but does not include defined mesh operations.

If you have modified the design setup, and do not want to use the existing current mesh, revert to the initial mesh before solving.

To revert to the initial mesh before solving:

- Click **Maxwell>Analysis Setup>Revert to Initial Mesh**.

Reverting to the initial mesh is useful when you want to evaluate how a different solution frequency affects the mesh generated during an adaptive analysis. You lose all solution data for a solve setup and all of its sweeps when you revert to the initial mesh for that setup. You can do this for all solve setups at once by selecting the command through Analysis in the menu system or project tree, or for a specific solve setup via its right mouse click menu in the project tree.

Related Topics

Plotting the Mesh

Technical Notes: The Mesh Generation Process

Generating the Mesh Without Solving

The mesh is automatically created when you run a solution analysis.

If you want to refine the mesh on a face or volume but do not want to generate a solution at the time, you can apply the mesh operations you have assigned without running the full analysis. Applying mesh operations without solving enables you to experiment with mesh refinement in specific problem regions without losing design solutions. You cannot undo the applied mesh operations, but you can discard them by closing the project without saving it.

To refine the mesh without solving, do one of the following after defining mesh operations:

- Click **Maxwell>Analysis Setup>Generate Mesh**.
- In the project tree, under **Analysis**, right-click the desired solution setup, and select **Generate Mesh**.

The same solve machine rules that apply to solving any other setup also apply here. The mesh operation will be sent to the default solve machine, or the Maxwell **Server Setup** dialog may appear to allow you to interactively specify a solve machine if "Prompt for analysis machine when launching analysis" is selected under **Tools>Options>General Options>Analysis Options** tab.

- If a current mesh has been generated, Maxwell refines it using the defined mesh operations.
- If a current mesh has not been generated, Maxwell applies the mesh operations to the initial mesh.
- If an initial mesh has not been generated, Maxwell generates it and applies the mesh operations to the initial mesh.
- If the defined mesh operations have been applied to the selected face or object, the current mesh is not altered.

Hint	Define a new mesh operation rather than modify an existing mesh operation. Maxwell does not re-apply a modified mesh operation.

Note	Mesh operations at the target design of a mu (permeability) link are ignored.

Related Topics

Plotting the Mesh

Technical Notes: The Mesh Generation Process

Setting HPC and Analysis Options

Remote Analysis

Specifying Initial Mesh Settings

You can specify the initial mesh settings, including the surface approximation and the meshing approach for both Maxwell 3D and Maxwell 2D designs. For most Maxwell 3D designs, you can let the solver automatically choose which of two meshing approaches to take. Maxwell predicts which one gives the best results, balancing mesh reliability, speed, quality, size and design

characteristics. In most cases, Maxwell uses **TAU** mesh, rather than the **Classic** mesh. In general, it looks for specific features (for example, stacks of large planar parallel facets with small gaps) and situations where the initial TAU mesh is four times larger than the Classic. For Maxwell 2D designs **TAU** mesh is the default.

In a few cases, you may decide to override the automatic or default choice and designate the mesher to use. To do so:

1. Select **Maxwell>Mesh>Initial Mesh Settings...** or in the Project tree, right-click on **Mesh**, and select **Initial Mesh Settings** from the shortcut menu. Alternatively, on the **Simulation** tab of the ribbon, select **Mesh Settings**.

   The **Initial Mesh Settings** dialog box appears with the **General** tab selected.

2. In the **Mesh Method** pane:

   For Maxwell 3D, you can select:

   - **Auto** (the default) – the solver automatically selects the mesher. In most cases, this will be **TAU**
   - **TAU** – only specific curve faces will be remeshe (for example, equation-based axisymmetric faces). If a curve face has connection to other non-remeshed curve faces, this curve face will not be remeshed. If the curve faces cannot be remeshed, faceting triangles will be used as the surface mesh. Mesh quality depends on how the faceting triangles are made. If the surface mesh is generated by TAU, this shows in the profile as Mesh TAU (Surface). If you check **Use alternative mesh methods as fall back** on the **Advanced** tab, this can fall back to Mesh TAU(Wrapper). This, in turn, can fall back to Mesh (stitch), which is classic.
   - **Classic** – this is based on the version 11 mesher. This setting must be used if you wish to use the Phi Mesher, for which you must also check **Allow Phi for layered geometry (Classic only)** on the **Advanced** tab.

   For Maxwell 2D, you can select:

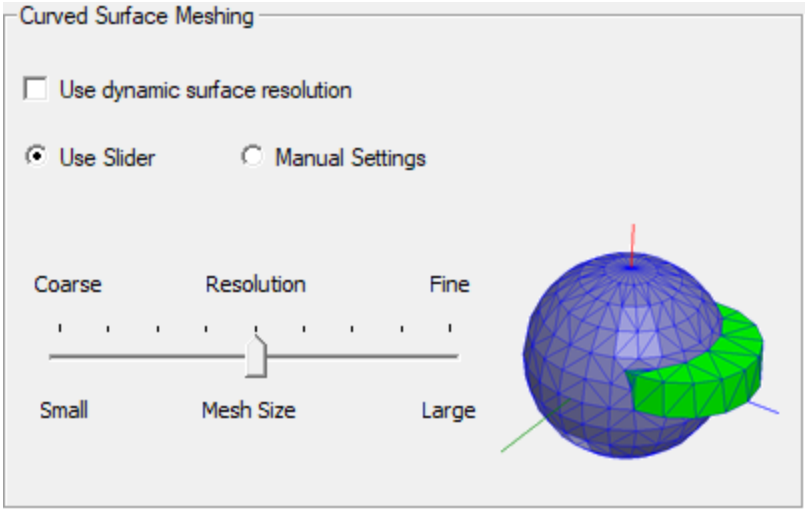
   - **TAU** (the default).
   - **Classic** – this is based on the version 11 mesher.

3. For Maxwell 3D only, you can also uncheck or check **Apply curvilinear meshing to all curved surfaces**. For models with curved surfaces, checking this increases accuracy, though it costs more memory.

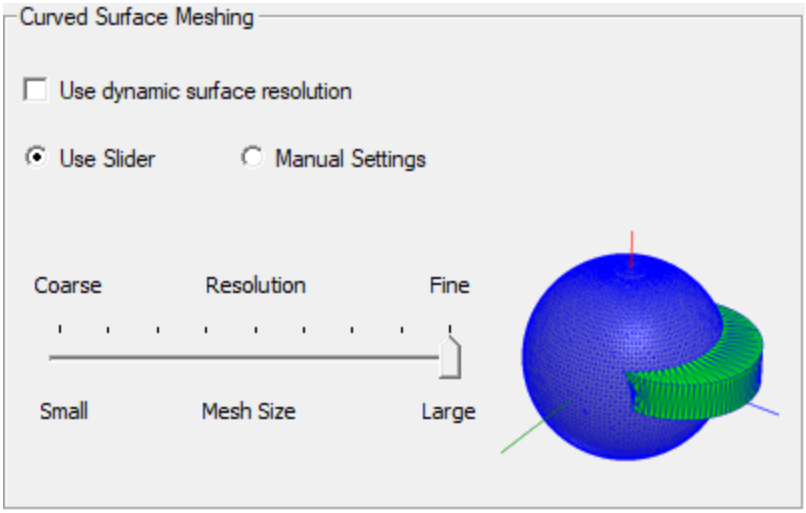
4. Under **Curved Surface Meshing**, you can chose to **Use dynamic surface resolution**, select **Use Slider** or specify **Manual Settings**.

   The slider includes a visual representation of the resolution you choose as ranging from **Coarse** Resolution with a **Small Mesh Size** through a nine position scale to a **Fine** Resolution with a **Large** mesh size.

   For example moving the slider to the left changes the figure as follows:

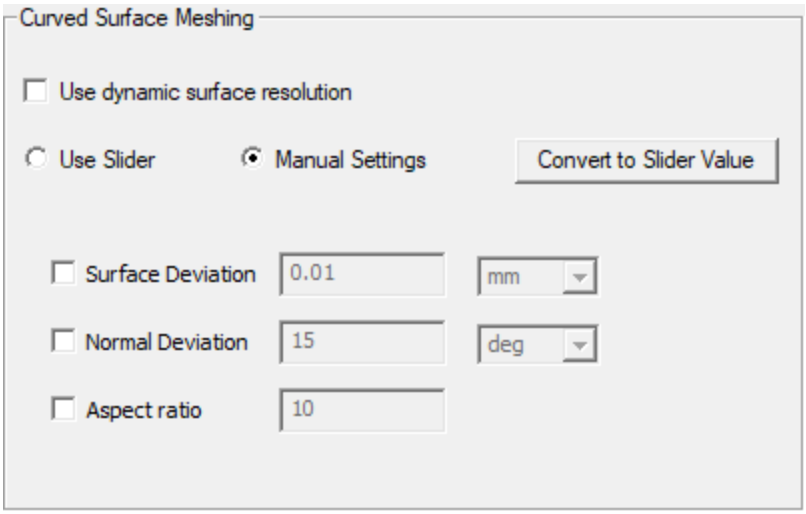


Moving the slider to the right changes the figure as follows:



5. Selecting **Use dynamic surface resolution** in the **Curved Surface Meshing** panel specifies the best-practice mesh operations over the geometric models. This mesh operation will support 3D volume mesh and surface mesh in all products. The default mesh operations or user defined mesh operations may or may not be replaced by the optimized mesh operations with model analysis. In general, large curve faces, curve faces with small gaps, or skewed cables get more smooth curvature representation while small curve faces, such as fillets, small curve objects, and so forth, get relatively coarse triangulations so that overall mesh size will be reduced.
   If you select dynamic surface resolution, you can optionally use the slider bar to adjust surface representation levels as: coarse (1-3), normal (4-6), and fine (7-9). You can also adjust the surface representation level for a specific group of faces by specifying slider bar, as described in Modifying Surface Approximation Settings.

6. If you choose **Manual Settings**, the dialog changes to show text fields.



Use the check boxes to enable the desired fields.

- For **Surface Deviation**, enter the distance between the true surfaces of the selected faces and the meshed faces in the text box.
- For **Normal Deviation**, enter the angular distance between the normal of the true surface and the corresponding mesh surface in the text box. Maxwell 3D's default normal deviation setting for the selected faces is 15 degrees. Maxwell 2D's default normal deviation setting is 3 degrees.
- For **Aspect ratio**, enter a value in the text box. This value determines the shape of the triangles. The higher the value, the thinner the triangles. Values close to 1 will result in well-formed, wide triangles. Maxwell's default aspect ratio settings for the selected faces are 10 for curved surfaces and 200 for planar surfaces.

The **Convert to Slider** button converts the manually entered values to an equivalent slider setting, and returns the panel to the slider view.

7. To make your choices the default, use the **Save as default** check box.
8. The **Advanced** tab lets you specify a **Set Length** for **Model Resolution**.

This setting is for experienced users who have a good understanding of how particular values will affect their models. In general, the **Auto** setting provides good results.

9. For Maxwell 3D models only, selecting the **Use Flex meshing for TAU volume mesh** feature enables a version of the TAU mesher that will rarely fail to generate a mesh. In most cases, the TAU Flex mesh is as accurate as traditional TAU or Classic meshes. However, for some complex models with bad translation or poorly defined surfaces that would fail to produce a strict mesh on all objects, relaxed tolerances will be applied. In these cases the user should review the mesh to evaluate whether it is acceptable for simulation. See *TAU Flex Meshing* for more details.

Note	TAU Flex meshing does not support RMxprt models.

10. **Use legacy faceter for TAU volume mesh** (both 2D and 3D models): By default, TAU uses the latest faceter to generate geometric model for meshing. Occasionally, the mesher may fail, and TAU will automatically fall back to use the legacy faceter to make a second attempt. In such cases, directly using the legacy faceter can get the mesh in the first attempt so total meshing time will be saved. The legacy faceter is not actively maintained by Spatial and will be retired soon. This option should be used only as a last resort.
11. For Maxwell 3D models only, selecting **Use alternative mesh methods as fall back** enables the **TAU** and **Classic** mesh methods (chosen on the **General** tab) to fall back to alternative mesh methods if the solver fails to generate the initial mesh.

    You can also select **Allow Phi for layered geometry (Classic only)**, which enables the **Classic** mesh method (chosen on the **General** tab) to use the Phi mesher for layered geometry.

    For example, the the sequence of mesher methods might go like this: When the mesh method is **TAU**: TAU(>Phi)>Classic. When mesh method is **Classic**: Classic(>Phi)>TAU. Note: the options do not affect the Auto mesh setups.
12. Click **OK** to apply your choices.

The settings will be applied to the initial mesh generated.

Related Topics

Defining Mesh Operations

TAU 2D Mesh Generation

TAU Flex Meshing

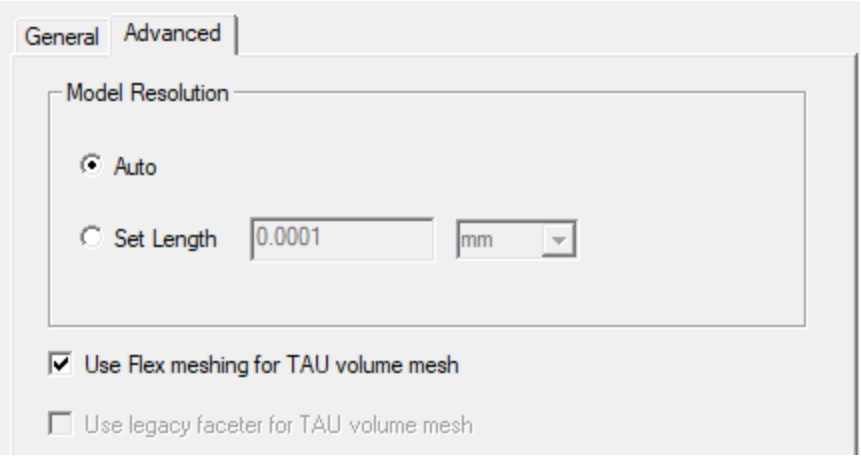
TAU Flex Meshing

TAU Flex is designed for guaranteed mesh success, and may also reduce meshing time. It provides a mesh even for complex and “dirty” geometry model without any preliminary geometric healing or repair.

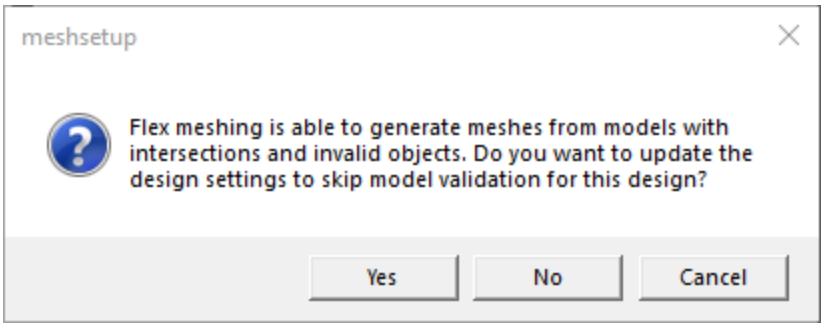
TAU Flex is a multi-domain, priority mesher which provides more efficient and reliable meshing. The priority mesh generation is driven by solver requirements for materials (metal/non-metal), boundary conditions (ports, radiation etc.), conduction paths, and so forth. The constraint tolerant meshing method gives the TAU Flex mesher flexibility to excuse mesh difficulties in lower-priority regions so that the resulting mesh can still provide acceptable simulation results.

Steps for Using TAU Flex

1. Navigate to the **Initial Mesh Settings** window.
2. Click the **Advanced** tab and select **Use Flex meshing for TAU volume mesh**.

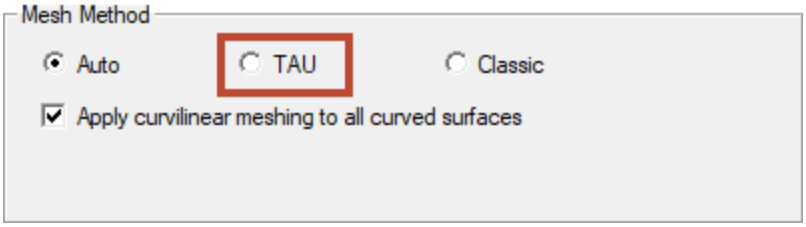


Selecting this option opens a dialogue. Click **Yes** to continue.

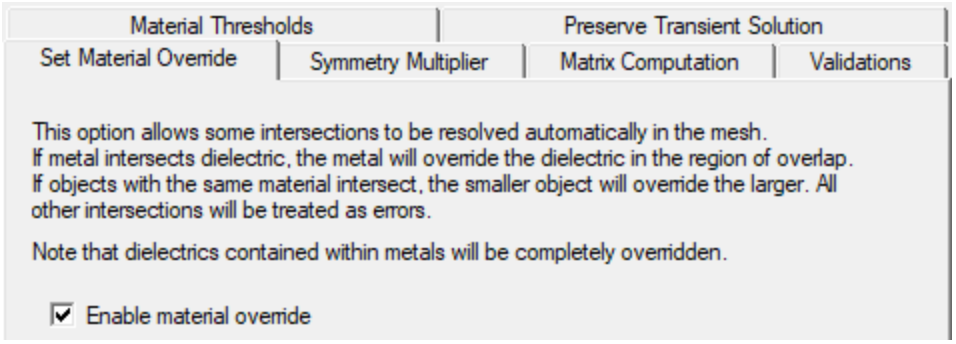


To get the full benefits of TAU Flex, especially for complex or “dirty” models, follow these recommended steps:

1. On the **General** tab of the **Initial Mesh Settings** dialog box, select **TAU**.

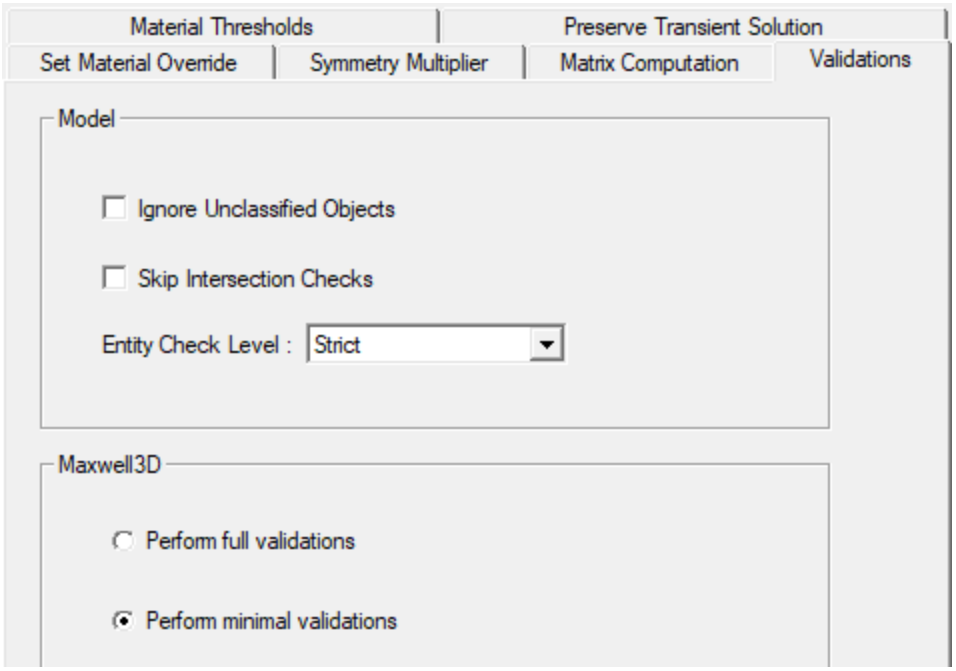


2. Click [*solver*]>**Design Settings** to open the **Design Settings** dialog.



   Check **Enable material override** in the **Set Material Override** tab to activate TAU's priority meshing method based on metal/non-metal materials.

4. Select the **Validations** tab of the **Design Settings** dialog and select **Perform minimal validations**.

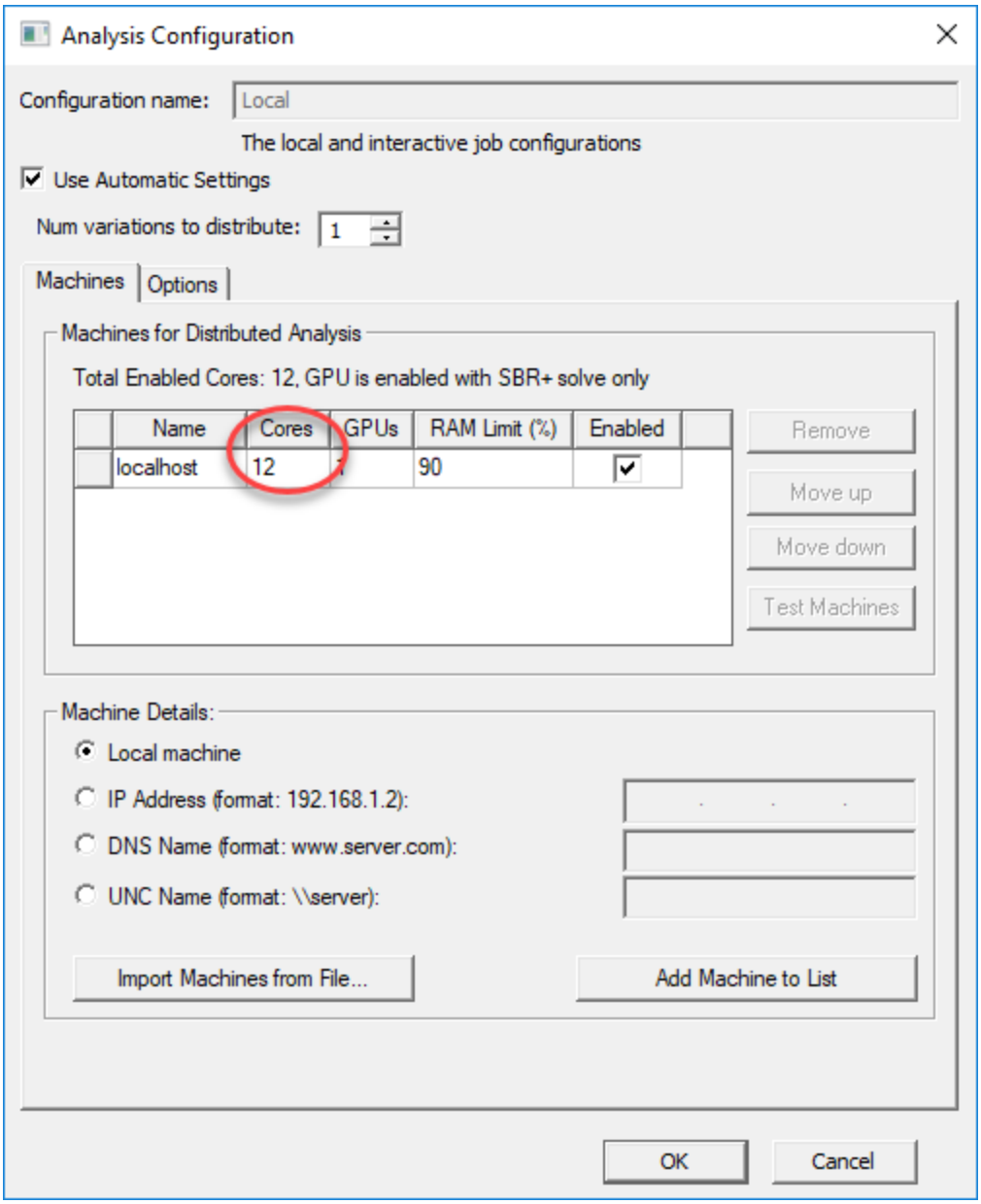


   This reduces pre-processing time for complex models, and allow simulation of "dirty" geometry for which TAU Flex can provide a mesh.

5. Because TAU Flex is a parallel mesher for many processes, you should specify multiple cores in **HPC and Analysis Options** to speed up the meshing process.Access the Analysis Configuration window one of two ways:

- Select the Simulation tab on the ribbon and click **Analysis Config**.
- Use **Tools** > **Options** > **HPC and Analysis Options** or select the **Simulation** tab on the ribbon and click **HPC Options**. Then double-click a configuration.

The **Analysis Configuration** window appears.



Specify the number of cores to use.

7. In most cases, TAU Flex can provide mesh as accurate as TAU or Classic, though not always. Therefore, we recommend that you **Generate Mesh** without solving; and that you review the mesh before proceeding with the simulation.

TAU Flex Warning Messages

In cases where TAU Flex has difficulty making a strict mesh on some objects, mesh warnings are given for those objects. You should review the mesh on those objects before proceeding to the simulation. You can exercise discretion in deciding whether a given mesh issue needs to be addressed (for example, critical antenna geometry) or can be ignored (for example, a small “leak” in a dielectric substrate far from the source).

There are three types of warnings, in decreasing order of severity: “Mesh Leak”, “Major Mesh Repair”, or “Minor Mesh Repair”.

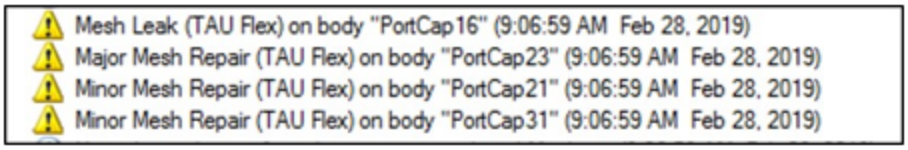
- **"Mesh Leak"** – *The most severe warning*. A body does not have any mesh element, or it has mesh elements with a significant mismatch of surface area between the mesh and geometry – larger than 10% of planar area, or 20% of curved area.
- **"Major Mesh Repair"** – A body has major surface area mismatch between the mesh and geometry – larger than 5% of planar area, or 10% of curved area.
- **"Minor Mesh Repair"** – *The least severe warning*. A body has minor surface area mismatch between the mesh and geometry – less than 5% for planar area, or 10% for curved area.

Reviewing TAU Flex Mesh Warnings

TAU offers three ways for you to view mesh warnings:

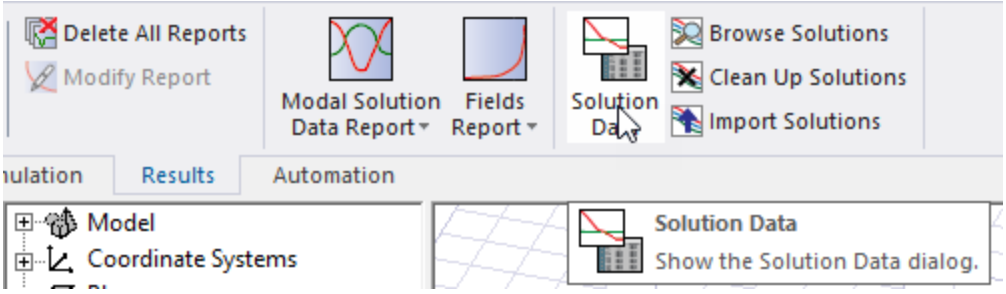
1. **Message Manager**:

   Immediately after the mesh generation is done, if a tolerant mesh is given, warnings will be displayed in the Message Manager of the Desktop. If the window is hidden, click **Show Messages**.

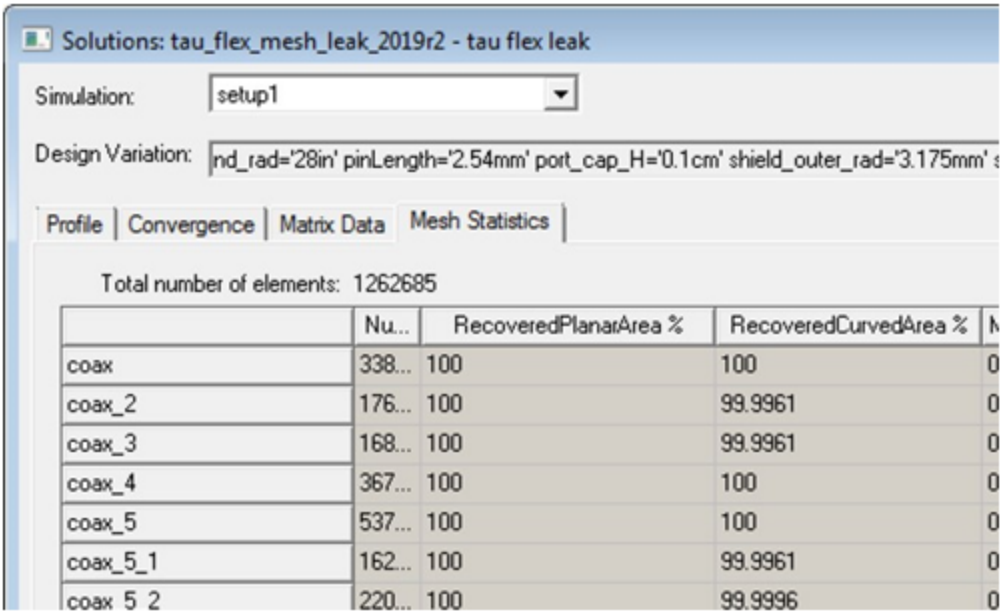


2. **Mesh Statistics**

   Use the **Solutions Data** icon in the **Simulation** tab of the ribbon to open a **Solutions** dialog box.

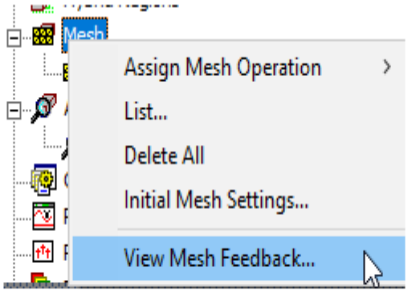


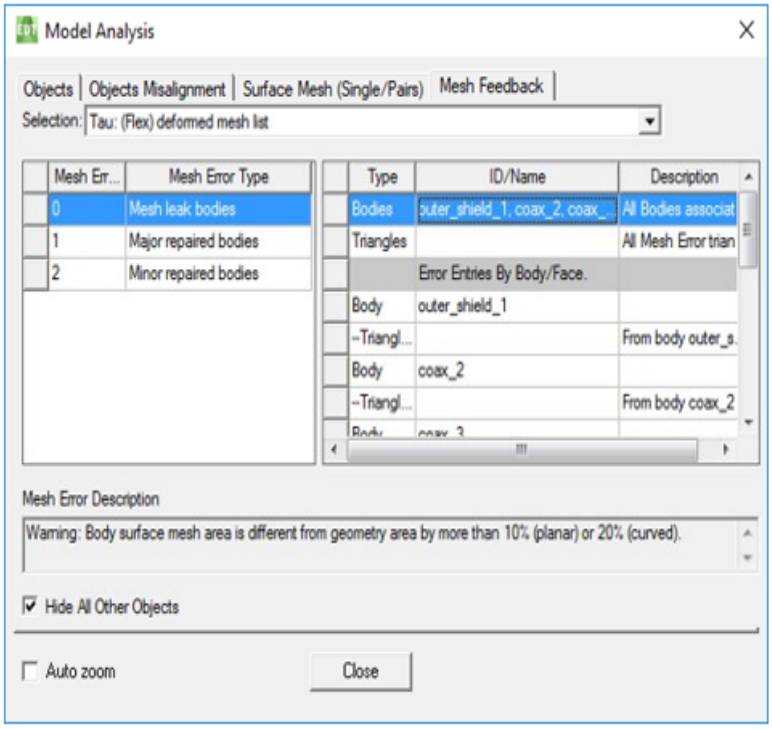
In the **Mesh Statistics** tab of the **Solutions** dialog box, if a tolerant mesh is generated, the surface area difference between the mesh and geometric model at body level is listed as “RecoveredPlanarArea” and “RecoveredCurveArea”. 100% means that the body has a strict mesh. You can click the surface area tab to sort the body list and see the tolerant mesh bodies first.



3. **Model Analysis dialog**, **Mesh Feedback** tab

   To view the details of the tolerant mesh, click **Modeler>Model Analysis>Show Analysis Dialog>View Mesh Feedback** to launch the **Model Analysis** dialog box.





Click on the top list to display objects that have tolerant mesh.

Click on the “Triangles” to display the mesh on the tolerant objects.

Individual objects will be listed under “Error Entities By Body/Face”. You can select one or multiple bodies to view the mesh at a more detailed level.

Addressing TAU Flex Mesh Issues

In cases where TAU Flex has difficulty making a strict mesh on some objects you can address the problem by using one of the following techniques:

1. Redraw the geometry in question in the native modeler
2. Heal the geometry either with the integrated healers in Maxwell or perhaps in SpaceClaim
3. If the objects has curved or true surfaces consider applying a localized surface mesh operation which creates a finer than default mesh.
4. Remove difficult objects that are not critical to the simulation.

Related Topics

Specifying Initial Mesh Settings

Generating the Mesh without Solving

Mesh Feedback Tab

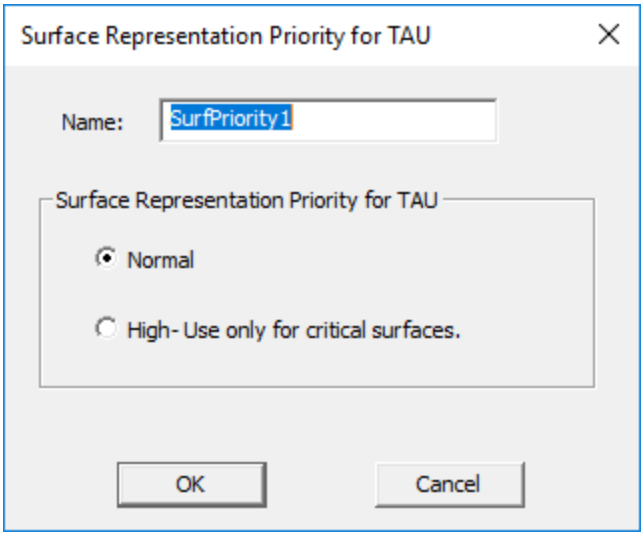
Specifying Surface Priority for TAU Mesher

You can choose to specify the surface representation priority for the TAU mesher. For most designs, you can let the solver automatically choose which of two meshing approaches to take. The solver predicts which one gives the best results, balancing mesh reliability, speed, quality, size and design characteristics. In most cases, the solver uses TAU mesh, rather than the classic mesh. In general, it looks for specific features (for example, stacks of large planar parallel facets with small gaps) and situations where the initial TAU mesh is 4 times larger than the Classic. Experienced users may choose to designate a high surface representation priority for the critical surfaces with the TAU mesh is used.

> **Important:**
>
> Use this mesh operation with caution. Use this option only if the model has tiny elements or critical boundary conditions since the solver performs several additional strict mesh verification steps on the selected objects.

1. Select the surface of interest.
2. If desired, specify a Name for the Mesh operation.
3. Right-click the **Mesh** icon in the Project tree to view the shortcut menu and select **Assign Mesh Operation** > **Surface Priority for Tau**.

   The **Surface Representation Priority for Tau** dialog box appears.



4. Select **High** for the critical surface.
5. Click **OK**.

   The Mesh operation appears in the Project Manager.

TAU 2D Mesh Generation

The Maxwell TAU (Triangular Adaptive Uniform) 2D mesher provides a fully-automated initial mesh generation method that creates a uniform, high quality, solver-driven mesh. It is designed for solvers without adaptive passes, or demanding high quality mesh. Several meshing methods are used by the TAU 2D mesher, such as skin depth layer mesh, clone mesh, rotational sweep mesh, and remeshing.

To benefit from these meshing technologies for Maxwell 2D designs, ensure that **TAU** is selected as the **Mesh Method** on the **Initial Mesh Settings** dialog **General** tab by right-clicking the Mesh icon in the Project Manager and selecting **Mesh>Initial Mesh Setting**, or by selecting **Maxwell>Mesh>Initial Mesh Settings...**. Note that **TAU** mesh is the default setting for new 2D designs.

When TAU 2D mesh is selected, its meshing methods are applied to specific local regions. The local regional meshes are then assembled as the final step for the simulation.

The following sections describe TAU 2D mesh generation methods for rotational models, for non-rotational models; and the Skin Depth Layer Mesh Generation method, which can be applied to any model.

Related Topics

TAU 2D Mesh Generation for Rotational Models

TAU 2D Regular Mesh Generation (Non-Rotational Models)

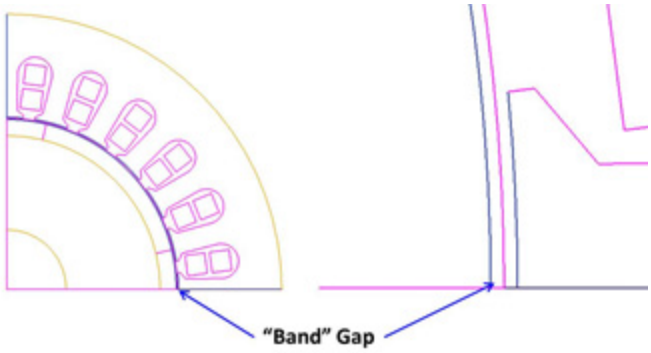
TAU 2D Skin Depth Layer Mesh Generation

TAU 2D Mesh Assembly

TAU 2D Mesh Generation for Rotational Models

If the mesher determines that a model is a rotational model, rotational mesh generation is used, in which the clone mesh and rotational sweep mesh methods described below are used.

To determine if the model is a rotational model, the curves around the origin are calculated. The smallest rotational gap is defined as the “Band” gap. The model shown below is an example of a rotational transient model.



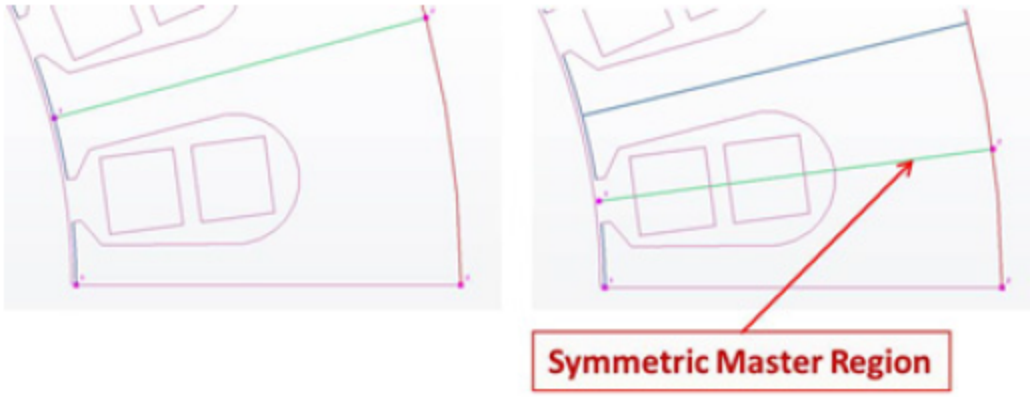
Two major regions: a static region and a moving region are classified by the Band gap.

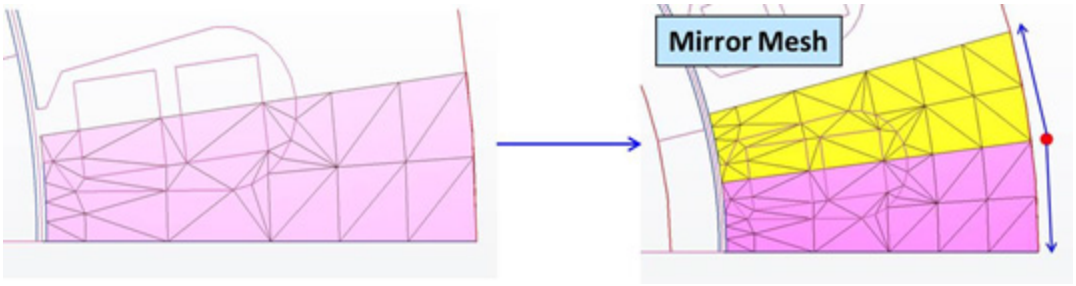
TAU 2D Clone Mesh Generation

Note	To use clone mesh, you must also setup Length-Based Mesh Refinement Inside Objects.

The static geometric model in the local rotational region is analyzed to determine whether it has multiple duplicate sub-regions. If a model has duplicate sub-regions, the TAU clone meshing is automatically activated to generate the mesh so that each sub-region has the topologically identical mesh. The following example shows a static clone region with six identical sub-regions - a master and clone regions.

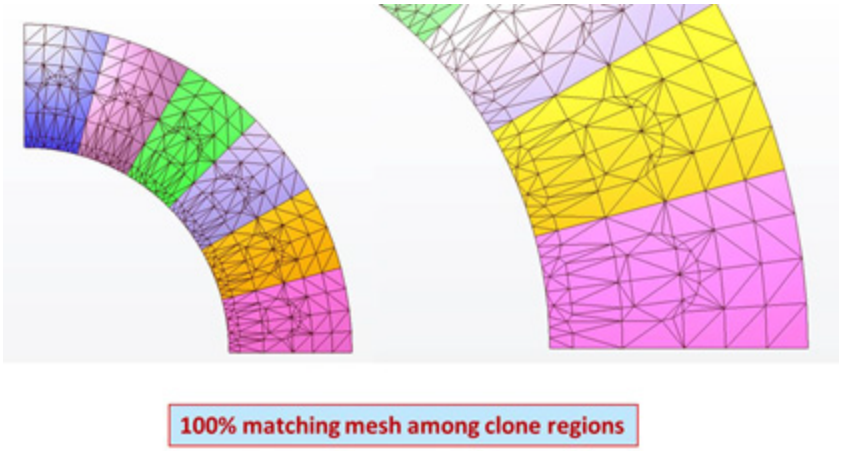
In the Clone region, the Master region is calculated and analyzed. If the Master region has the symmetric feature, the symmetric mesh is generated.



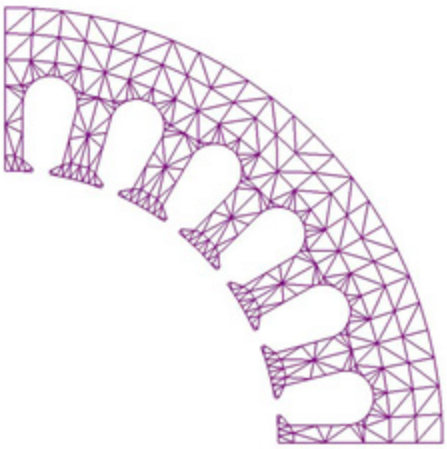


As shown above, a uniform mesh will be generated in one half of the Master region first, then the mesh is mirrored about the symmetric central line to generate the symmetric mesh in the Master region.

Because all the child regions have identical geometric features, the mesher uses copy-paste to replicate the Master mesh at each of the child regions, rotating it to the appropriate angles.



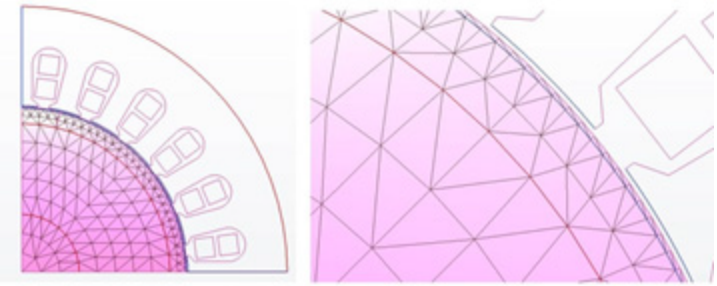
This ensures the identical mesh in the different regions, especially for the teeth.



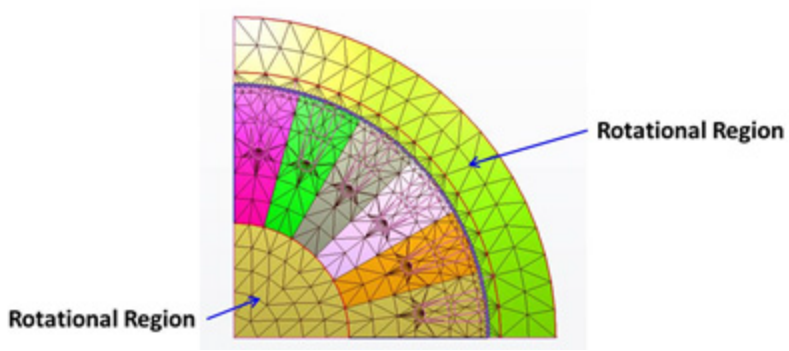
TAU 2D Rotational Sweep Mesh Generation

If the local region does *not* have identical geometric sub-regions and thus the clone mesh is *not* generated, then geometric analysis is performed to verify whether the region is rotational.

Around the rotational center, if a region has two or more rotational curves, this region can be defined as a “rotational” region. In this case, “Rotational sweep” mesh generation is used proceeding in the radial direction with layers while using uniform sweep elements in the rotational direction. This meshing method controls the aspect ratio in the radial direction, while maintaining the uniform sweep in the rotational direction.



Rotational Sweeping Mesh



Related Topics

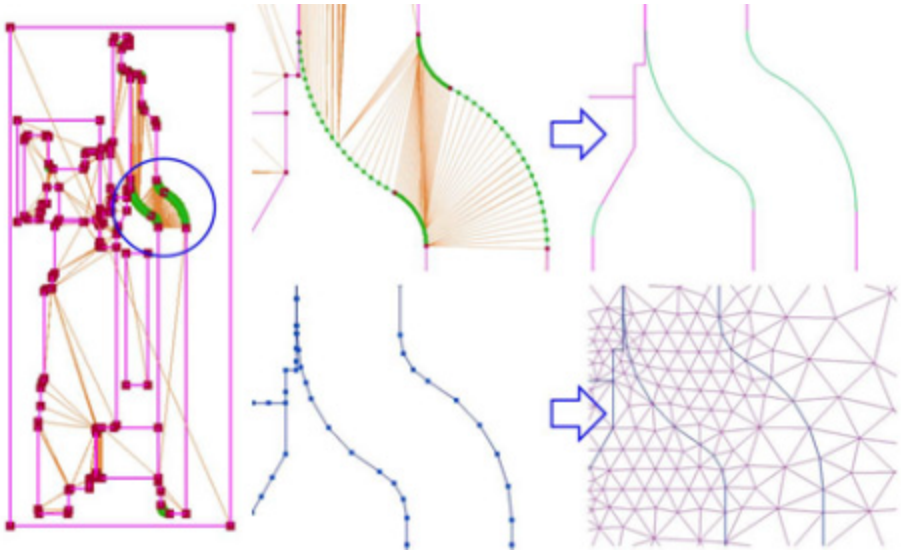
TAU 2D Regular Mesh Generation (Non-Rotational Models)

TAU 2D Skin Depth Layer Mesh Generation

TAU 2D Mesh Assembly

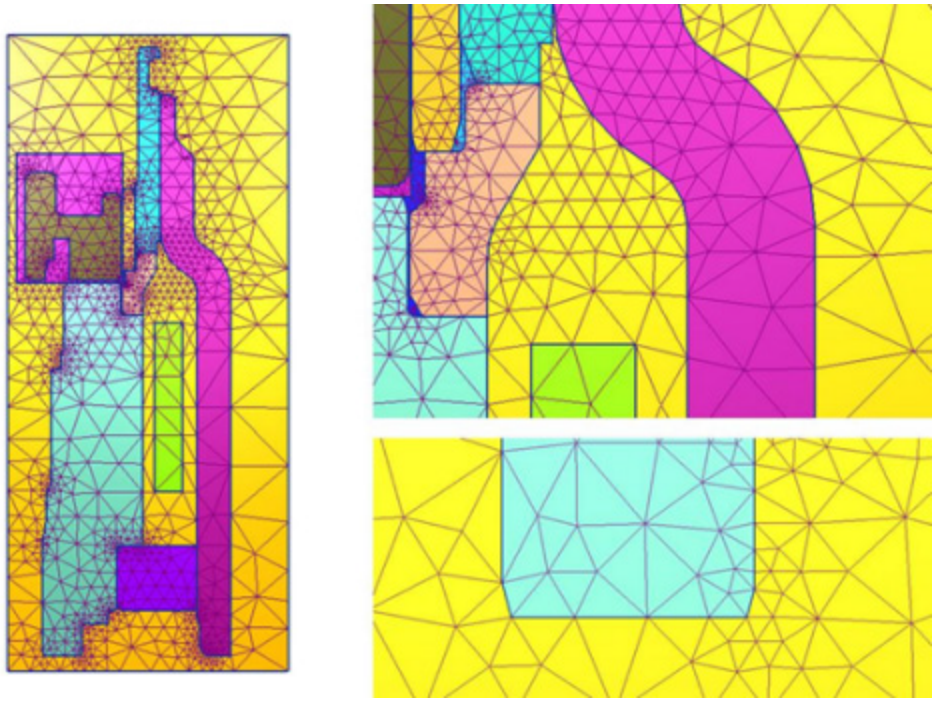
TAU 2D Regular Mesh Generation (Non-Rotational Models)

If the model is not identified as a rotational model, the TAU 2D regular mesher is used to generate the mesh. TAU 2D mesher does not use Parasolid union as the Classic mesher does. Instead it reads the objects individually as-is. Remesh technology is used to get the best representation of the curves with minimum mesh points.



The TAU 2D regular mesher uses a top down meshing algorithm that starts the mesh from a uniform base mesh, then adapts the mesh in the necessary regions, and does the boundary recovery later. Mesh quality is controlled and maintained during the entire meshing process. To

maintain a uniform mesh, 1:2 splitting method is used for the transitions from small elements to large elements.



In most cases, users do not need to add any additional seed refinement mesh operations. The mesh can be directly used to do the transient simulation. For solvers with adaptive passes, a few passes can be saved for the convergence although the initial mesh is larger since it is high quality and uniform.

Related Topics

TAU 2D Mesh Generation for Rotational Models

TAU 2D Skin Depth Layer Mesh Generation

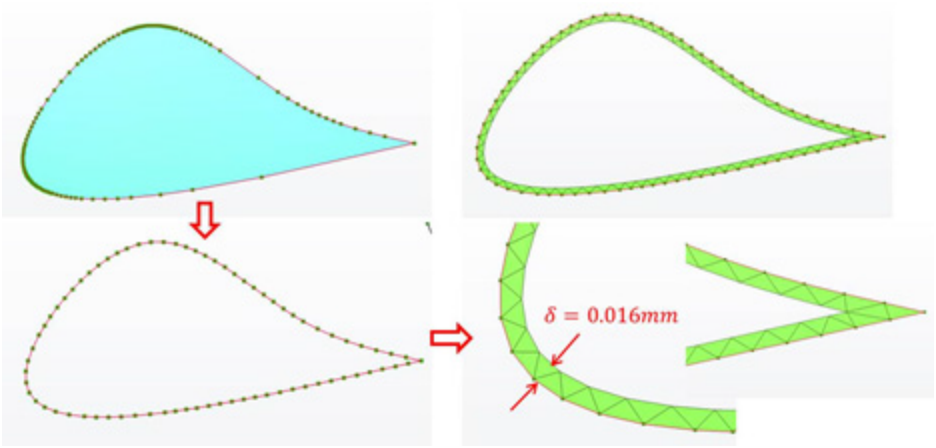
TAU 2D Mesh Assembly

TAU 2D Skin Depth Layer Mesh Generation

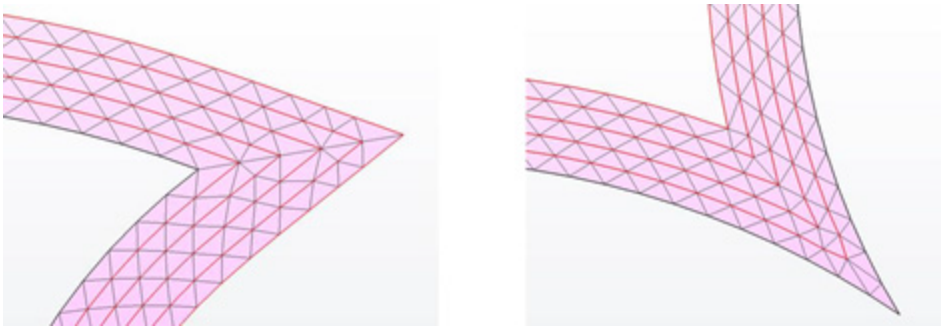
For this mesh generation method, the number of layers and the thickness can be specified by selecting one or more body edges. They will be exactly respected in the TAU mesh unless the layer thickness is too large, or layers from other directions overlap, intersect, or leave a narrow gap.

The layer mesh starts from the specified edges, and advances toward the direction of the body region layer-by-layer. If two bodies contact each other, the edges from these two bodies at the contact region may coincide, *appearing* as one edge even though there actually are two or more edges. Therefore, please be careful when picking the edges in multiple body contact regions. In

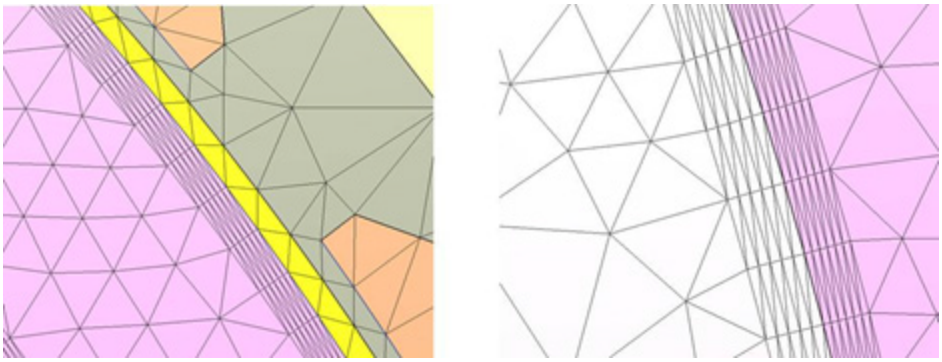
such cases, it is recommended that you hide all other bodies and only display the desired body, then make the edge selections. Remesh technology is then used to generate the mesh on the edges based on the thickness of the first layer to maintain the mesh quality.



The aspect ratio in the layer mesh is defined by the element length in the tangential and normal directions. The ratio will be in the range of 1 to 40 based on the smallest thickness in the layers. Different layer mesh methods are using depending on the ratios. If the ratio is smaller than 4.0, the middle points of the edges are used to advance layers forward as shown below.



If the ratio is more than 4.0, edge points are used to advance layers.



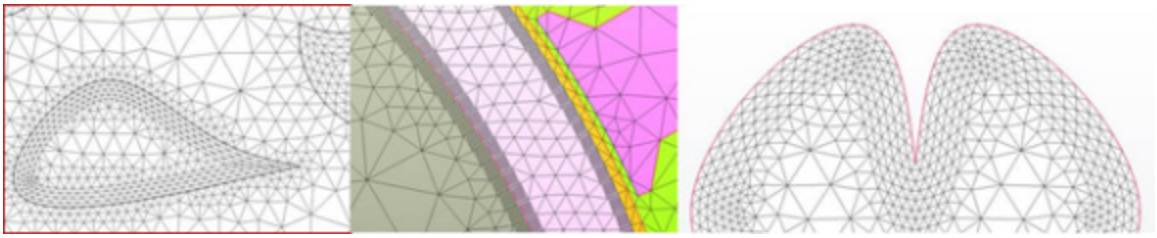
Messages are displayed if the thickness is too small or too large.

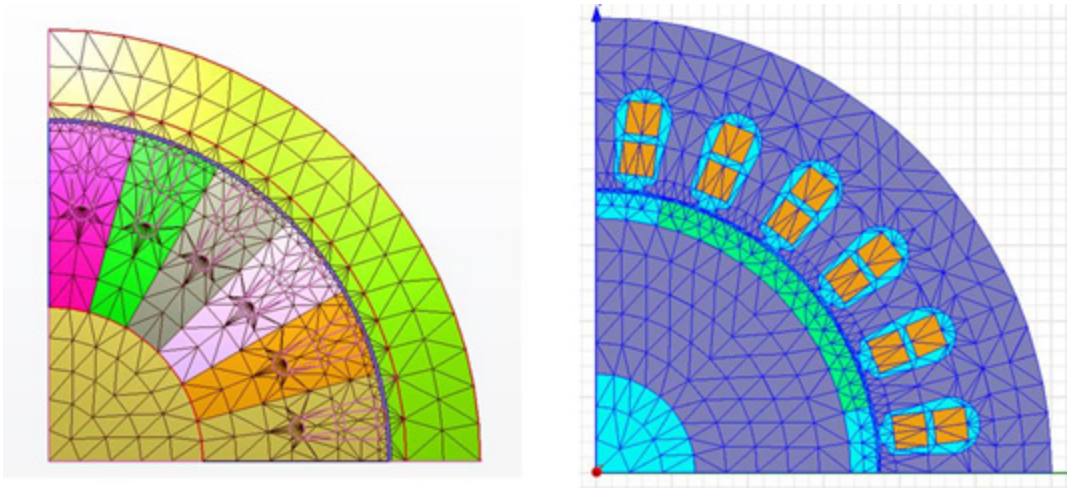
Related Topics

Skin Depth Layers Setting for 2D
TAU 2D Mesh Generation for Rotational Models
TAU 2D Regular Mesh Generation (Non-Rotational Models)
TAU 2D Mesh Assembly

TAU 2D Mesh Assembly

The final step to get a conformal mesh for the entire model, assembles the local region meshes generated by the clone mesh, rotational sweep mesh, and skin depth layer mesh methods with the mesh generated by the TAU 2D regular meshing method as the final step.





Related Topics

TAU 2D Mesh Generation for Rotational Models
TAU 2D Regular Mesh Generation (Non-Rotational Models)
TAU 2D Skin Depth Layer Mesh Generation

Layer Mesh Generation for 3D Rotational Models

Layer Mesh generation for 3D rotational models can only be applied to non-skewed bodies in Maxwell rotational models in which the CylindricalGap mesh operation is used.

- Only body level selection is allowed.
- Rotational Layer mesh can only be generated on non-skewed objects for current release.
- User can select multiple bodies, and assign a layer mesh operation to all of them.
- Rotational layer mesh is only generated for bodies near the band.
- For selected bodies, no layer mesh will be generated in the axial direction, or for matching boundaries.

There are three types of layer meshes for 3D rotational models.

- Edge Cut Layer Mesh
- Rotational Layer Mesh
- Edge Cut + Rotational Layer Mesh

These mesh operations will work only if the clone mesh works.

Related Topics

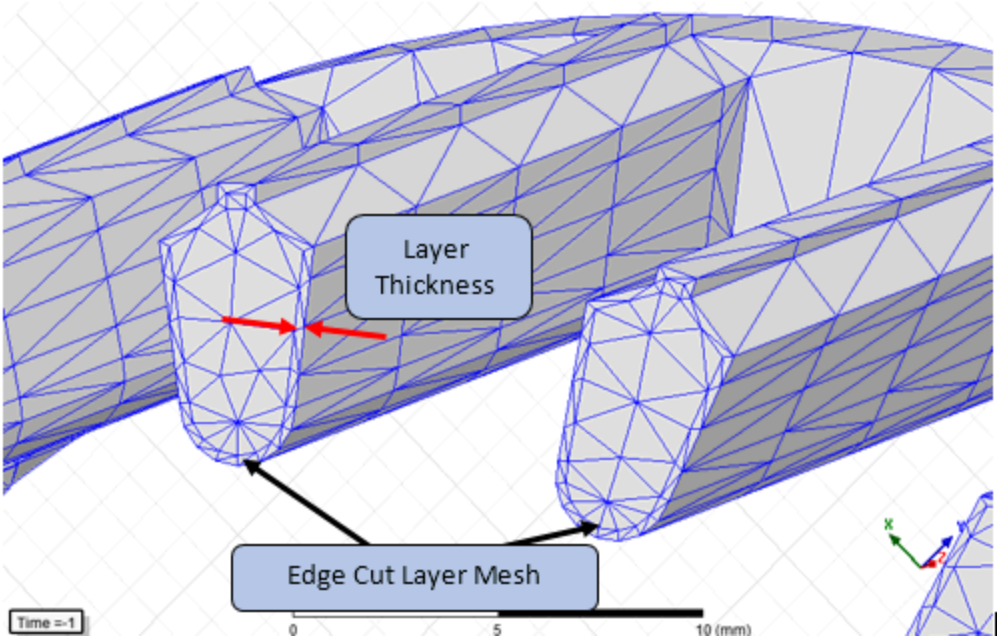
"Assigning Cylindrical Gap Treatment " on page 15-12

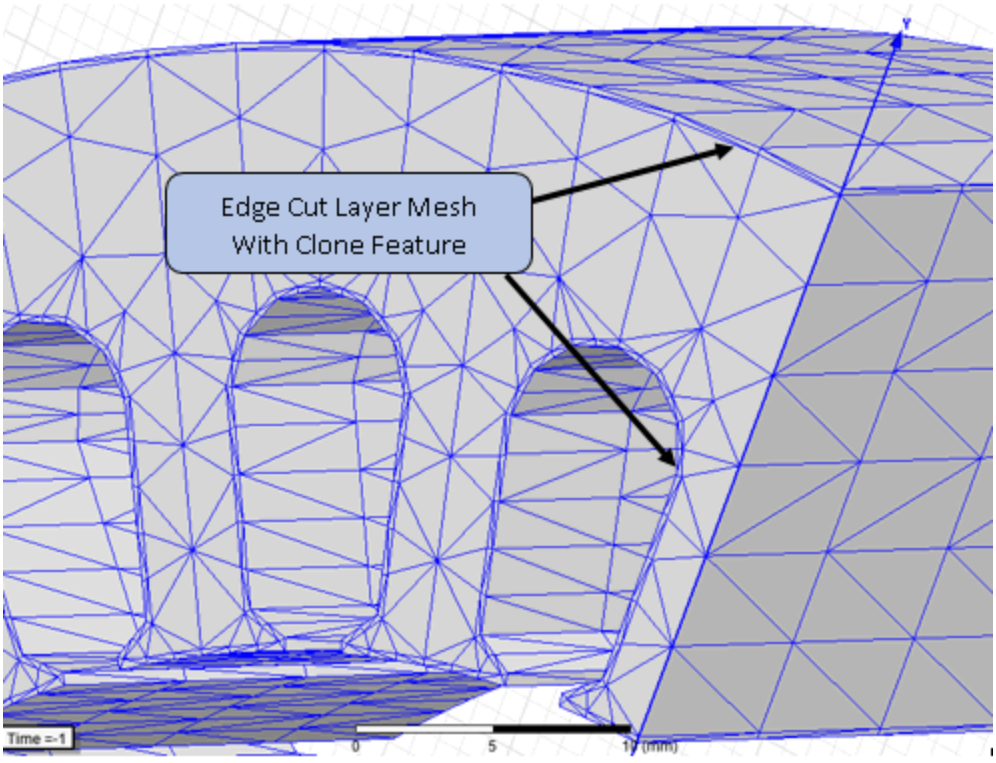
"3D Clone Mesh Generation" on page 15-13

"Changing Properties and Reassigning Mesh Operations" on page 15-69

Edge Cut Layer Mesh Generation

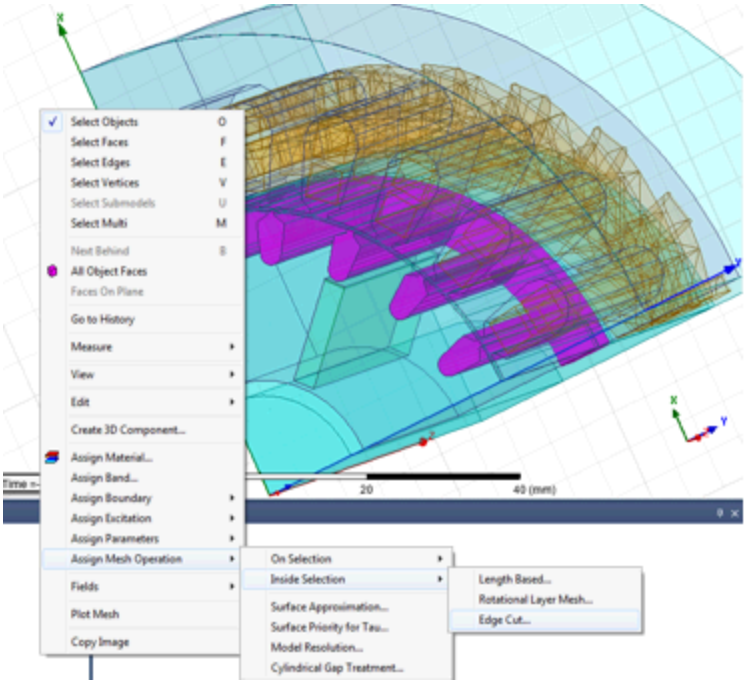
Edge Cut Layer Mesh creates one layer of mesh on the bodies with the layer thickness defined by user input. Layer mesh will not be created at the top or bottom of the axial direction and matching boundary faces. If the object has clone feature, the layer mesh will be coupled with the clone mesh.



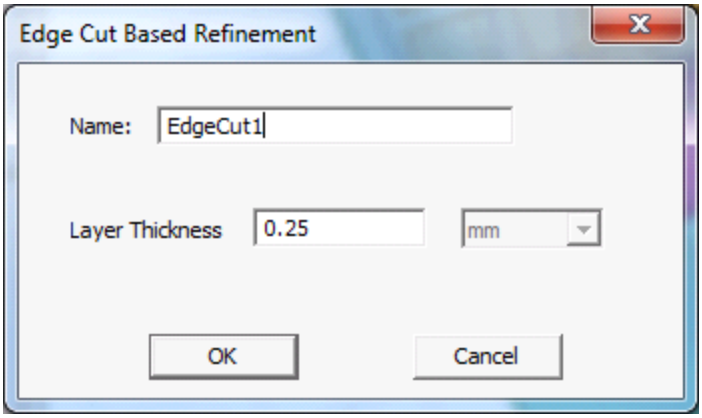


To apply Edge Cut Layer Mesh:

1. Select the desired bodies near the band.
2. Right click to open the context menu, and select **Assign Mesh Operation>Inside Selection>Edge Cut Based**.



3. In the **Edge Cut Based Refinement** dialog box, you can rename the mesh operation, and specify the layer thickness. The recommended layer thickness is the band gap distance.



4. Click **OK**.

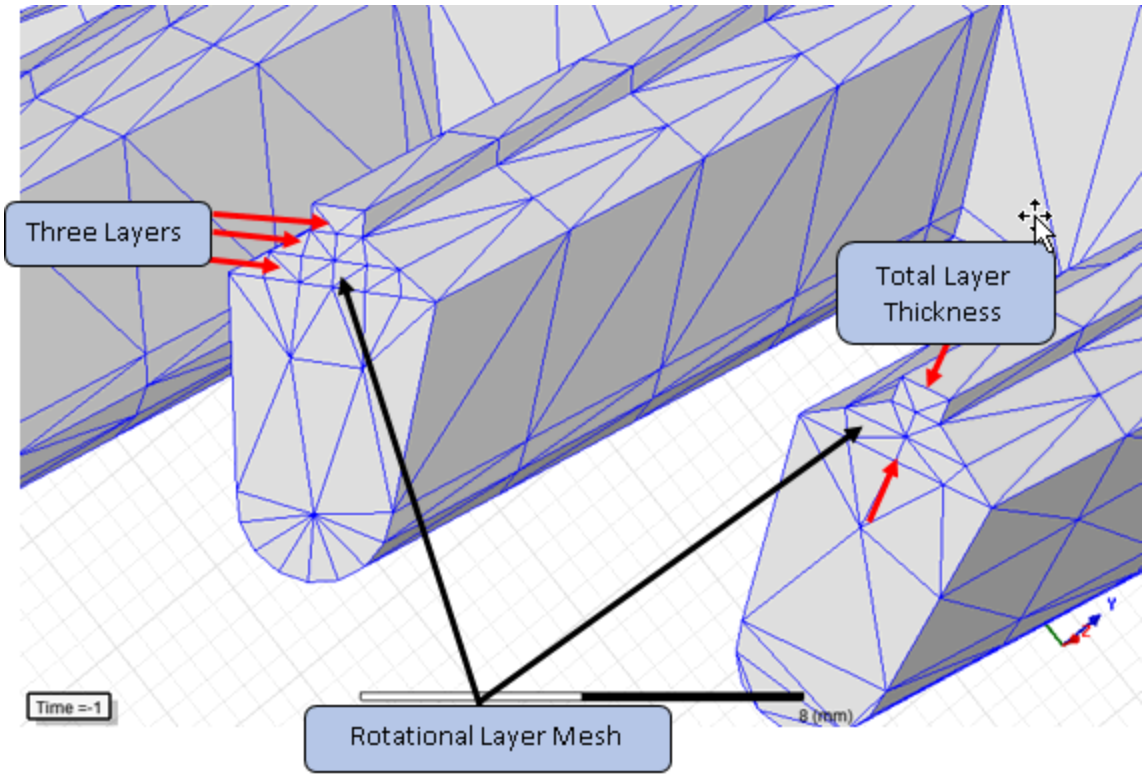
Related Topics

"Layer Mesh Generation for 3D Rotational Models " on page 15-63
"Rotational Layer Mesh Generation" on the next page

"Edge Cut + Rotational Layer Mesh Generation" on page 15-68
"Changing Properties and Reassigning Mesh Operations" on page 15-69

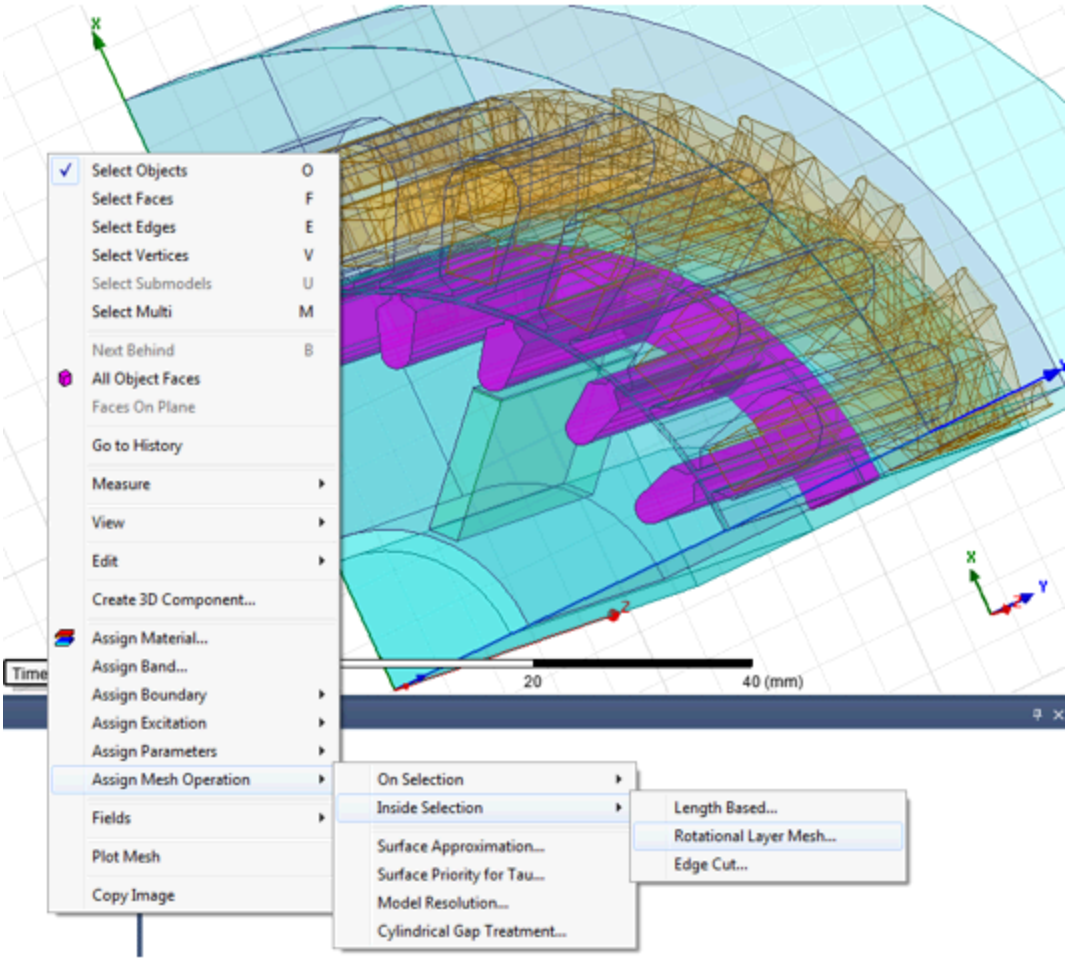
Rotational Layer Mesh Generation

Rotational Layer Mesh creates up to three layers in the radial direction starting from the faces most adjacent to the “Band”. You can define the **Number of Layers** (up to three) and **Total Layer Thickness**.

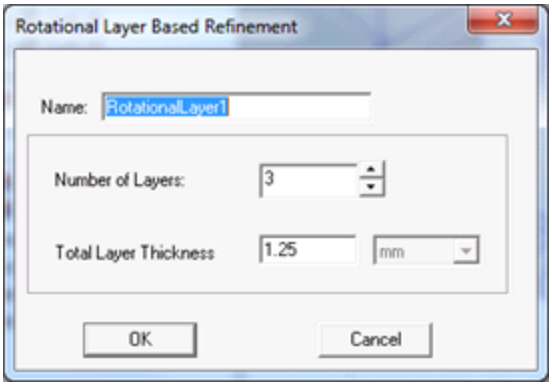


To apply Rotational Layer Mesh:

1. Select the desired bodies near the band.
2. Right click to open the context menu, and select **Assign Mesh Operation>Inside Selection>Rotational Layer Mesh**.



3. In the **Rotational Layer Based Refinement** dialog box, you can rename the mesh operation, specify the number of layers (maximum of 3), and the total thickness of all the layers combined.



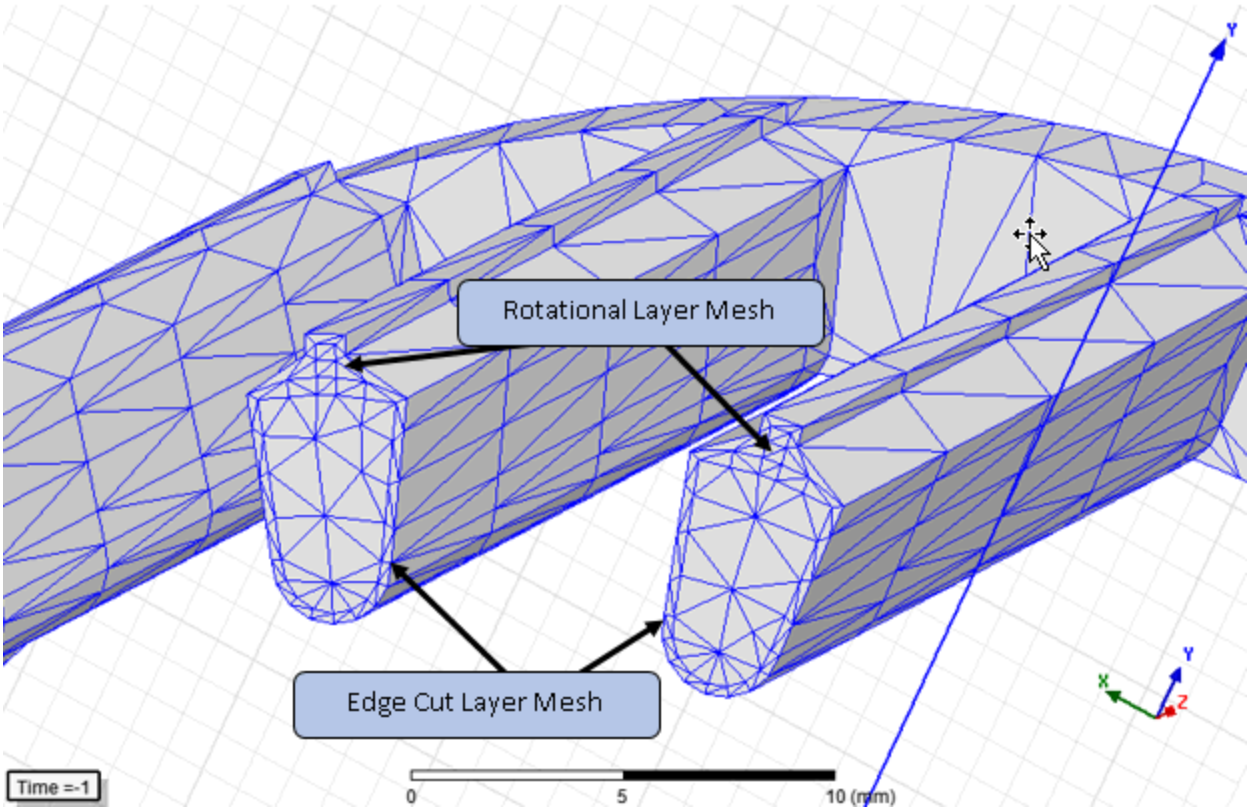
4. Click **OK**.

Related Topics

"Layer Mesh Generation for 3D Rotational Models " on page 15-63
"Edge Cut Layer Mesh Generation" on page 15-63
"Edge Cut + Rotational Layer Mesh Generation" below
"Changing Properties and Reassigning Mesh Operations" on the facing page

Edge Cut + Rotational Layer Mesh Generation

These settings are a combination of Edge Cut Layer Mesh and Rotational Layer Mesh settings, and as such has the properties of both. One layer of edge cut mesh will be generated first, then extra multiple layers will be added by the rotational layer mesh settings.



To apply Edge Cut + Rotational Layer Mesh:

1. Select the desired bodies near the band.
2. Right-click to open the context menu, and select **Assign Mesh Operation>Inside Selection>Edge Cut**.
3. In the **Edge Cut Based Refinement** dialog box, you can rename the mesh operation, and specify the layer thickness. The recommended layer thickness is the band gap distance.
4. Reselect the same bodies.

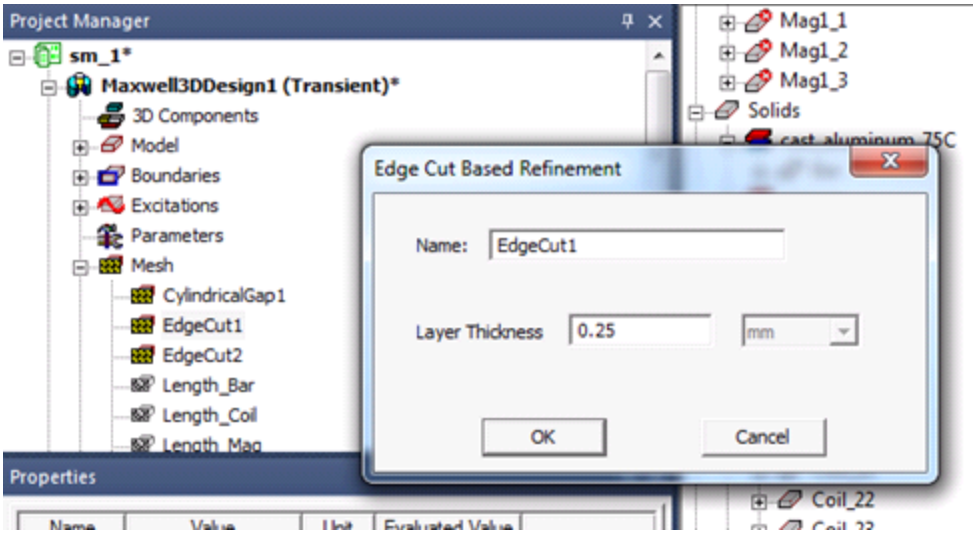
5. Right-click to open the context menu, and select **Assign Mesh Operation>Inside Selection>Rotational Layer Mesh**.
6. In the **Rotational Layer Based Refinement** dialog box, you can rename the mesh operation, specify the number of layers (maximum of 3), and the total thickness of all the layers combined.
7. Click **OK**.

Related Topics

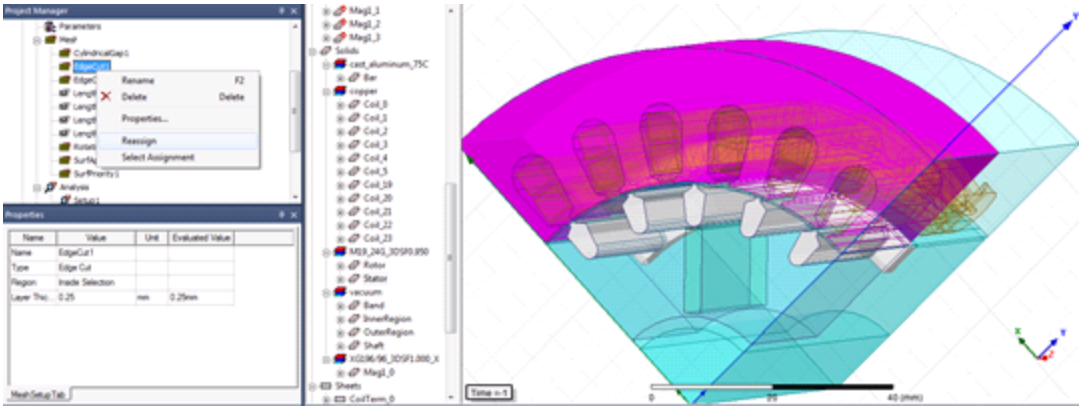
"Layer Mesh Generation for 3D Rotational Models " on page 15-63
"Edge Cut Layer Mesh Generation" on page 15-63
"Rotational Layer Mesh Generation" on page 15-66
"Changing Properties and Reassigning Mesh Operations" below

Changing Properties and Reassigning Mesh Operations

To alter the mesh settings, find the entry under Mesh in the Project Manager. Double click to bring up the dialog box. Change the layer thickness or number of layers as necessary.



To reassign which bodies the mesh operations are applied to, first select all the bodies you would like to assign the bodies to. Then right click the mesh operation in the Project Manager and select “Reassign.” The mesh operation will now be reassigned to the selected objects.



Related Topics

"Edge Cut Layer Mesh Generation" on page 15-63
"Assigning Cylindrical Gap Treatment " on page 15-12
"Edge Cut + Rotational Layer Mesh Generation" on page 15-68
"Layer Mesh Generation for 3D Rotational Models " on page 15-63

Deleting All Previously-Assigned Mesh Refinements

To delete previously-assigned mesh refinement data:

- Click **Maxwell>Mesh>Delete All**.

Note	This will not alter any existing meshes that have already made use of one or more mesh refinement operations. However, deleted mesh operations will not be used again after reverting to the initial mesh.

Viewing List of Mesh Operations

To list all mesh operations for the project, do the following from the project tree:

1. Right-click **Mesh**.
   A shortcut menu appears.
2. Select **List** from the shortcut menu.
   The **Design List** dialog box appears, with the **Mesh Operations** tab displayed.
3. Click **Done** to close this dialog box.

Reassigning a Mesh Operation

To reassign a previously-assigned mesh setup, do the following in the project tree:

1. Select the object or objects to which you want to reassign the mesh operation.
2. Under **Mesh**, right-click the mesh refinement you had previously completed.
   A shortcut menu appears.
3. Select **Reassign** from the shortcut menu.
   The mesh operation is reassigned to the currently selected objects.

To view the objects to which a mesh operation is assigned:

1. In the Project tree, select the mesh operation.
2. Right-click the mesh refinement you had previously completed.
   A shortcut menu appears.
3. Select **Select Assignment** from the shortcut menu.
   The objects to which the operation is assigned are selected.

Related Topics

Defining Mesh Operations

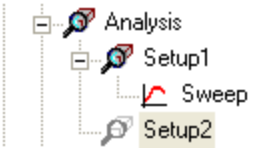
Viewing Mesh Properties

To view mesh properties, do the following in the project tree:

1. Under **Mesh**, right-click the mesh refinement you had previously completed.
   A shortcut menu appears.
2. Select **Properties** from the shortcut menu.
   The dialog box for the type of mesh operation associated with the selected mesh refinement appears.

16 - Specifying Solution Settings

Specify how Maxwell computes a solution by adding a *solution setup* to the design. You can define more than one solution setup per design.



Each solution setup includes:

- General data about the solution's generation.
- Adaptive mesh refinement parameters if you want the mesh to be refined iteratively in areas of highest error.

To add a solution setup to a design:

1. Select a design in the project tree.
2. Click **Maxwell3D**>**Analysis Setup>Add Solution Setup** or **Maxwell2D**>**Analysis Setup>Add Solution Setup**.
   - Alternatively, right click **Analysis** in the project tree, and then click **Add Solution Setup** on the shortcut menu.

   The **Solve Setup** dialog box appears containing several tabs.

   For Non-Transient Solutions

   The following tabs are present for non-transient solutions:

General	Includes general solution settings. Also contains a button for setting **HPC and Analysis Options**, which allows you to select or create an analysis configuration.
Convergence	Includes settings for convergence.
Expression Cache	Contains an optional list of expressions that you can specify for convergence criteria.
Solver	Includes settings for nonlinear residual and matrix solver type.
Frequency Sweep	Includes setting for varying the frequency of an Eddy Current or AC Conduction solution. **NOTE:** This tab is present only for Eddy Current and AC Conduction solutions.
Defaults	Enables you to save the current settings as the defaults for future solution setups or to revert the current settings to Maxwell's standard settings.

   For Transient Solutions

   The following tabs are present for transient solutions:

General	Includes general solution settings. Also contains a button for setting **HPC and Analysis Options**, which allows you to select or create an analysis configuration.
Save Fields	Enables you to create a time point list for saving field solutions.
Advanced	Supports user control program, import mesh options, and starting/continuing from a previously solved setup.
Solver	Includes settings for nonlinear residual solver type, for enabling the solver to fast reach steady-state, and for selecting the Time Decomposition Method.
Expression Cache	Contains an optional list of expressions that you can specify for evaluating solutions.
Defaults	Enables you to save the current settings as the defaults for future solution setups or to revert the current settings to Maxwell's standard settings.

For Electric Transient Solutions

The following tabs are present for electric transient solutions:

General	Includes general solution settings. Also contains a button for setting **HPC and Analysis Options**, which allows you to select or create an analysis configuration.
Solver	Includes settings for initial condition, temporal tolerance, computing power loss data, and importing a mesh.
Expression Cache	Contains an optional list of expressions that you can specify for evaluating solutions.
Defaults	Enables you to save the current settings as the defaults for future solution setups or to revert the current settings to Maxwell's standard settings.

3. Specify the desired settings on the tabs.
4. If you want to disable an individual analysis, uncheck the **Enabled** check box. Refer to *Disabling and Enabling an Analysis Setup* for additional information.
5. If you want to use the default values, click **Use Default**.
6. Click **OK**.

Related Topics

Analysis Parameters

Setting Analysis Parameters for Non-Transient Solutions

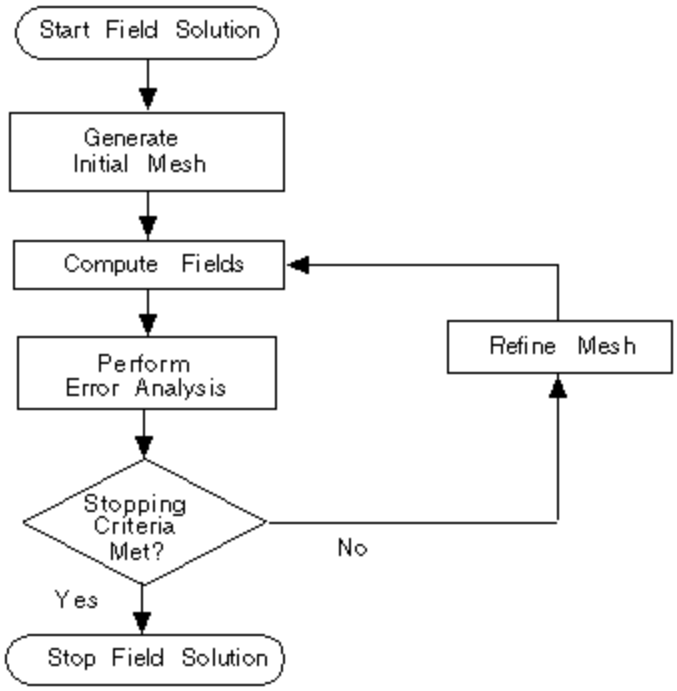
Setting Analysis Parameters for Transient Solutions

Setting Analysis Parameters for Electric Transient Solutions

Disabling and Enabling an Analysis Setup
Resetting Analysis to Time Zero
Changing Memory Settings
Specifying the Solver Type
Technical Notes: *Meshing Aspects for 3D Transient Applications With Motion*
Defining Mesh Operations
Copying a Solution Setup

Analysis Parameters

During adaptive analysis the system iteratively refines the starting mesh in order to reduce the size of individual elements in areas of high error — thus improving the accuracy of the solution.



When an adaptive analysis is performed:

- Maxwell generates a field solution using the specified mesh.
- It then analyzes the accuracy of the solution by calculating an energy value based on the error in the solution. The exact mechanism for evaluating the error varies by solution type. For example, in magnetostatic it can use Curl H to find the current density and then subtract all input currents and other sources. For a perfect solution the result would be zero, for a real, finite mesh the result is some amount of residual current density. An energy value calculated from this residual current density is called the error energy. The “Energy Error %” is the error energy as a percentage of the total energy (calculated with the original sources).
- If more than 1 pass has been completed, the software also calculates the change in total energy from the previous pass. The percentage difference is the “Delta Energy (%)”. Adaptive refinement continues until both the “Energy Error %” and the “Delta Energy (%)” are below the target Percent Error specified by the user (or until it reaches the Maximum Number of Passes requested).
- When the error targets are not satisfied the mesh is refined. This is generally done by subdividing the elements with the highest error energy into smaller elements.
- The user can optionally request to “Use Output Variable Convergence”. This is an additional stopping criterion. The Energy Error % and the Delta Energy must still be below the target Percent Error, but the software will also compute the specified Output Variable for each adaptive pass and will calculate the percentage change in that value for each pass after the first (this is the “Output Var. Delta (%)”). The solution will continue until the energy error criteria are met and the Output Var. Delta is below the target “Max. Delta Per Pass” specified by the user for output variable convergence (or until Maximum Number of Passes).

Related Topics

Setting Analysis Parameters for Non-Transient Solutions

Setting Analysis Parameters for Transient Solutions

Setting Analysis Parameters for Non-Transient Solutions

When you set up an analysis, define the parameters on the following tabs of the **Solve Setup** dialog box (which appears when you click **Maxwell**>**Analysis Setup>Add Solution Setup**):

- General (Name, Adaptive Setup, Parameters)
- Convergence
- Expression Cache
- Solver
- Frequency Sweep (Eddy Current and AC Conduction solutions only)
- Defaults

Defining Settings on the General Tab for Non-Transient Solutions

The following settings can be defined on the **General** tab of the **Solve Setup** dialog box for non-transient solutions:

- Setup Name
- Adaptive Setup
- Parameters

Setup Name for Non-Transient Solutions

By default, **Setup1** appears as the name of the first analysis you set up.

To change the name of the setup, type the new name in the **Name** text box on the **General** tab of the **Solve Setup** dialog box.

To disable the setup, uncheck the **Enabled** check box. Refer to *Disabling and Enabling an Analysis Setup* for additional information.

Adaptive Setup for Non-Transient Solutions

To specify adaptive settings, click the **General** tab in the **Solve Setup** dialog box and use the options in the **Adaptive Setup** section. Adaptive settings include the following:

- **Maximum Number of Passes.** The **Maximum Number of Passes** is the maximum number of mesh refinement cycles you would like Maxwell to perform. This value is a stopping criterion for the adaptive solution; if the maximum number of passes has been completed, the adaptive analysis stops. If the maximum number of passes has not been completed, the adaptive analysis continues unless the convergence criteria are reached.

Note	The size of the finite element mesh — and the amount of memory required to generate a solution — increases with each adaptive refinement of the mesh. Setting the maximum number of passes too high can result in Maxwell requesting more memory than is available or taking excessive time to compute solutions.

- **Percent Error**. The **Percent Error** allows you to control the desired solution accuracy. Smaller values produce more accurate (but slower) solutions; larger values produce less accurate (but faster) solutions. At each step in the adaptive process, the energy and error energy are computed, and you can view the most recent solutions as soon as they are completed. After the mesh is refined, the matrix is calculated on the refined mesh. The relative change between the previous matrix and the current matrix is then computed and reported as the matrix delta; the target matrix delta is the **Percent Error**. A small delta indicates that further mesh refinement will probably not change the solution.

Parameter Settings

On the **General** tab of the **Solve Setup** dialog box, if you want to solve for fields only and no other parameters, then click the **Solve Fields Only** check box.

When this check box is cleared, the following options are available:

- **Solve Matrix**. Select **After last pass** or **Only after converging**.

Defining Settings on the Convergence Tab for Non-Transient Solutions

To set the convergence data, click the **Convergence** tab in the **Solve Setup** dialog box. Convergence settings can be standard or optional. Standard convergence settings include the following:

- **Refinement Per Pass** – The percent you set for **Refinement Per Pass** determines how many tetrahedra are added at each iteration of the adaptive refinement process. The tetrahedra with the highest error are refined.
- **Minimum Number of Passes** – The minimum number of adaptive passes even if the convergence criteria are reaches.
- **Minimum Converged Passes –** The minimum number of adaptive passes to continue after the convergence criteria are reached.

Optionally, convergence criteria can also be based on the value of an output variable. This condition is in addition to the normal stopping criteria. To base the convergence criteria on an output variable value, refer to Defining Expressions for Non-Transient Solutions.

For 2D Eddy Current solutions only, improved accuracy may be obtained by including loss in the convergence criteria. To include the loss calculation in convergence, select the **Use Loss Convergence** check box.

Note	Maxwell 2D versions prior to version 12 did not use loss in the convergence calculation.

Related Topics

Defining Expressions for Non-Transient Solutions

Defining Expressions for Non-Transient Solutions

You can specify additional convergence criteria through the use of expressions and output variables. The **Maximum Delta** or the **Maximum Percent Delta** defined for expression convergence represents the difference in values of the expressions between consecutive adaptive passes. If the difference in the value of the expression between consecutive passes is less than the **Maximum Delta** or the **Maximum Percent Delta** value this part of the convergence criteria is satisfied.

To set expressions as convergence criteria:

1. Right-click on the Analysis icon in the Project tree to open the **Solution Setup** dialog box.
2. Click the **Expression Cache** tab and click the **Add...** button to open the **Add to Expression Cache** dialog box.
3. Specify the Context for the expression, including **Report Type**, **Solution**, **Parameter**.
4. Under the **Trace** tab, select from the **Category**, **Quantity**, and **Function** lists to create expressions.

   Selecting a listed category lists the Quantities and Functions available for each category. If you have defined one or more output variables, you can see them by selecting the **Output**

Variables category. The **Output Variables** button opens a dialog box that lets you define output variables. See Specifying Output Variables.

When you have created an expression, it appears in the **Expression** field of the **Trace** tab. If desired, you can use the **Range Function** button to select range functions to apply to the expression.

Under the **Calculation Range** tab, you can view the values of available sweep variables. Clicking the ellipsis **[...]** button in the **Edit** column opens a list of values from which you can select.

5. When finished defining the expression that you want to add to the cache, click the **Add Calculation** button.

   This adds the selected expression and the associated context to a table in the **Expression Cache** tab. You can define additional expressions with contexts and add them in the same way.

6. When finished adding expressions, click **Done** to close the **Add to Expression Cache** dialog box.

   The **Expression Cache** tab lists the expressions you have added in a table.

   - The **Title** field is editable, by default showing the name as built from the expression (with underscores removed).
   - The **Expression** field shows the full expression. If necessary, you can resize the **Solution Setup** dialog box. You can also resize each columns in the table.
   - The **Context** column shows None for Modal solutions, or the appropriate geometry for Fields calculations.
   - The **Intrinsics** column shows a clickable button that opens an **Edit Calculation Range** dialog box. If the **Intrinsics** column button shows **None** you cannot edit the value. If the button shows variables, click the ellipsis **[...]** in the **Edit** column to display a list of the variable values from which you can select. Click **OK** to close the **Edit Calculation Range** dialog box and apply your selections to the **Expression Cache**.

7. To designate one or more expressions for convergence, click the field for the **Convergence** column for each expression.

   This opens an **Adaptive Convergence** dialog box. Check **Use this expression for convergence** to enable the radio buttons. You can then specify the **Maximum Delta** (an absolute change in value between passes) or the **Maximum Percent Delta** (a percentage change in value between passes) criteria. The **Maximum Delta** provides a way around a potential issue if your expression is essentially zero and the numeric noise from pass to pass causes the maximum percentage delta to remain high. In that case adaptive refinement continues until you get to the maximum number of passes.

8. Optionally, buttons allow you to **Edit** and **Remove** selected expressions, and to **Remove All** expressions in the table.

9. For Eddy Current only check boxes allow you to save the expression cache either with **Evaluate Cache vs. Pass**, **Evaluate Cache vs. Freq**, or both.

10. When finished defining expressions, click **OK** to close the **Solution Setup** dialog box.

Related Topics

Viewing Convergence Data
Viewing the Output Variable Convergence
Specifying Output Variables

Defining Settings on the Solver Tab for Non-Transient Solutions

The settings on the **Solver** tab of the **Solve Setup** dialog box depend on the solution type selected for that project.

- Magnetostatic Solver Settings
- Eddy Current Solver Settings
- Electrostatic Solver Settings
- AC Conduction Solver Settings
- DC Conduction Solver Settings

Magnetostatic Solver Settings

For magnetostatic solutions, you can define the following settings on the **Solver** tab of the **Solve Setup** dialog box:

- Enter a value in the **Nonlinear Residual** text box.
- For both 2D and 3D magnetostatic designs, you can select the **Smooth BH Curve** check box. This allows the solver to "smoothen" the list of points to make a refined BH curve for better solving. If this option is not selected, the solver just uses the discrete points.
- For 3D magnetostatic designs only, you can Enable Iterative Solver.
- For both 2D and 3D magnetostatic designs, you can select the **Number of Nonlinear Iterations** check box. This provides an option for you to set both a **Minimum** iteration number to avoid non-converged solution, and a **Maximum** iteration number to avoid taking too much computation time. These values must be integers greater than 0. By default the values for **Minimum** and **Maximum** are 1 and 100, respectively. Maximum number should be greater than the Minimum number.
- **Advanced Option** settings allow you to perform more sophisticated simulations of the magnetostatic field.
    - **Use pre-computed permeability data** (for general soft magnetic materials) uses a previously solved mu (apparent permeability) distribution from a linked source design. Selecting this option enables the **Setup Link** button, and automatically opens the **Setup Link** dialog in which you set up the link. Optionally, you can override the solved mu with a user-specified program, which you can specify on the **Advance** tab of the **Setup Link** dialog box. Select the **Including magnets** check box if magnets are included in the distribution. Refer to "Permeability Options for Magnetostatic Solutions " on the facing page for details on using these settings.
    - The **Import mesh** check box can be selected in the target design if the mesh for the same geometry in the linked source design is to be used in the analysis. Selecting

this option enables the **Setup Link** button, and automatically opens the **Setup Link** dialog in which you set up the link. Refer to "Import Mesh for Non-Transient Solutions " on page 16-20 for details on using this setting.

- If an advanced option has previously been set up, you can click the **Setup Link** button to open the **Setup Link** dialog and edit the link settings.

Note	The data link used to link source and target designs uses data caching technology. Therefore, if the design changes, to make sure the new data (from the source design) is used in the target design, right-click the **Analysis** field of the target design in the project tree, and select **Clear Linked Data**. This flushes existing data.

Related Topics

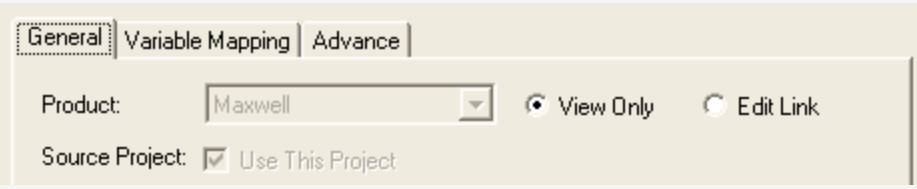
"Enabling the Iterative Solver" on page 16-24
"Import Mesh for Non-Transient Solutions " on page 16-20
"Permeability Options for Magnetostatic Solutions " below
"Setup Link for Non-Transient Solutions" on page 16-22

Permeability Options for Magnetostatic Solutions

To set permeability options in the **Solve Setup** dialog box:

1. Click **Maxwell**>**Analysis Setup>Add Solution Setup** to open the **Solve Setup** dialog box.
2. Click the **Solver** tab.
3. In the **Advanced Option** section, select **Use pre-computed permeability data**, which uses a previously solved mu (apparent permeability) distribution from a linked source design. Selecting this option enables the **Setup Link** button, and the **Setup Link** dialog opens, allowing you to specify the source project, design, and solution containing the desired information. The mesh between the two linked designs is the same. You can use the apparent permeability that has been frozen from a previously solved source design.

   By default, the **Setup Link** dialog opens on the **General** tab with only the **This project** radio button selected.

Note	If a link had previously been set up, the **Setup Link** dialog opens in **View Only** mode with all settings disabled. 

	To enable settings, you can select **Edit Link**.

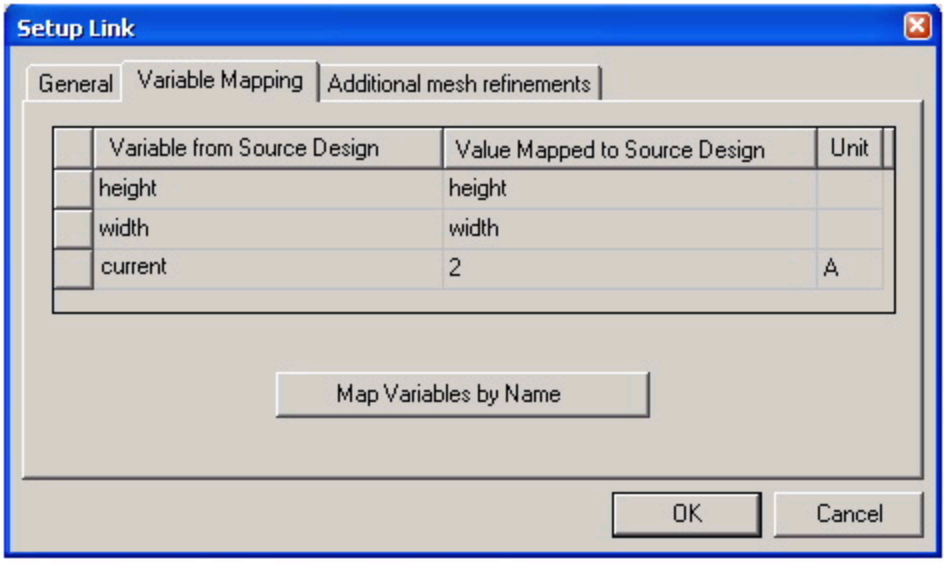
4. To select a source project do one of the following:
   - To use the current project as the source, check **Use This Project**. This disables the **Save source path relative to:** radio buttons and the ellipsis **[...]** button and its associated text field.
   - To specify a source project file other than the current project click the ellipsis **[...]** button to open a file browser window. When you have selected the project, click the **Open** button to accept the project file for the setup. You can use the check box to **Open as read only**.

   Use the radio buttons to specify whether to save the source path relative to **The project directory of selected product** or **This project**.
5. When you select a source project file, the **Source Design** and the **Source Solution** fields are filled in with default values, and their drop down menus contain any available designs and solutions. You can use the drop down menus to choose from the available designs and solutions.

   The “Default” solution is the product-dependent solution of the first Setup – the setup listed first in the source design's project tree (alphanumerical order). A product-specific solution of this setup becomes the default solution. In most designs, it is LastAdaptive.
6. Use the check box to specify whether to **Simulate source design as needed**.
7. Use the check box to specify whether to **Preserve source design solution**. Note that in the extractor mode, the source project will be saved upon exit. Extractor mode means that the software is opened during the link solely for the purpose of solving.
8. The **Variable Mapping** tab lets you view any variables contained in the selected Project.

   When there are variables in the source design, you can choose to “map” these variables to constant values, expressions or variables in the target designs. Variable mapping becomes more important when the datalink type requires source and target design to be geometrically identical and source design is geometrically parameterized.

   For linked designs with variables of the same name, you can click **Map Variables by Name** to automatically map same named variables. In this example the variables “height” and “weight” are mapped to the Source, whereas the “current” variable does not have a same named counterpart, and retains its value in the source design.



9. Optionally, you can override the solved mu with a user-specified control program, which you can specify on the **Advance** tab of the **Setup Link** dialog box. Additionally, because the non-linear characteristics of many magnetic materials are too complex to be defined through the definition of a BH-Curve, an advanced feature allows the permeability characteristic of the target design to be obtained by mathematical manipulation of the permeability from the source design via a user-defined control program. To use this feature:

    a. Check **Enable user program**.
    b. Specify the **Absolute path of program** to the user control program.
    c. Specify any **Additional arguments** needed by the user control program.

Note	For detailed information on using this feature, refer to *Using the Advanced Permeability Option*.

10. Click **OK** to accept the setup and close the Setup Link dialog and return to the Solve Setup dialog.
11. After setting up the link, you can select the **Including magnets** check box if magnets are included in the distribution.
12. When you are finished specifying solution settings, click **OK** to close the **Solve Setup** dialog box and finalize the solution setup.

Note	The usual way to create the linked designs is by copying and pasting an existing design from the project of your choice once the design is complete. Occasionally an existing design (possibly from a different project) can be used as a source. The data link automatically attempts to solve the linked (source) design if necessary (if no solution exists), then automatically copies the mesh and necessary data (calculated distribution of permeability), and finally performs the requested calculation in the target design. This data link allows you to calculate

	the magnetic field to be performed in a design with, for example, modified sources while keeping the permeability in a "frozen" state calculated in a different design (the same geometry and mesh but with different sources).

Using the Advanced Permeability Option

The non-linear characteristics of many magnetic materials are too complex to be defined through the definition of a BH-Curve. This feature enhances the existing permeability-link by enabling the permeability characteristic of the target design to be obtained by mathematical manipulation of the permeability from the source design. The nonlinear “apparent” permeability from the source design is frozen (linearized) and used by the target design. Because the permeability in the target design is linear, the apparent and the incremental permeabilities will be identical and the apparent and incremental inductances will also match.

This manipulation is done via a user-supplied control program. A user-specified executable is launched at the beginning of target design simulation and the linked permeability data (i.e., the “frozen” mu) from the source design is overridden by the values calculated by the user control program. The target design solver is then launched and uses the overridden value in its calculation.

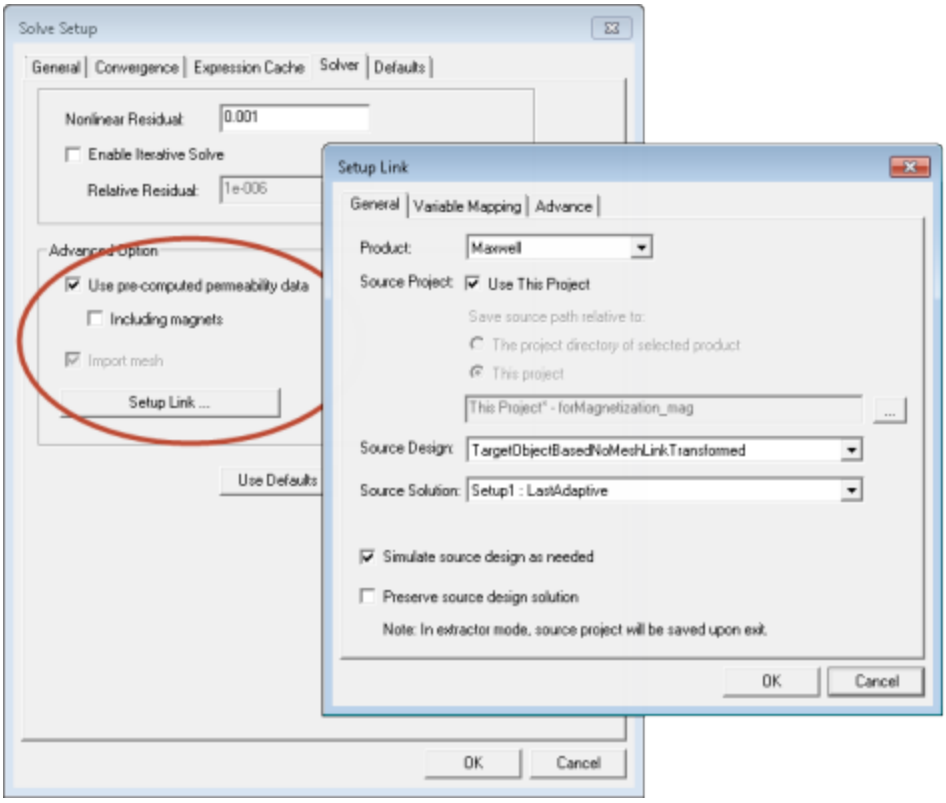
The user control executable program is launched with two file names as arguments:

1. **Per mesh element permeability value.** This is the solved mu (i.e., the apparent permeability not the incremental permeability) from the source design. Note that the format of this file is different between 2D and 3D designs.
2. **Per mesh element information.** Information includes the centroid location (x, y, z), Bx, By, Bz at the centroid and Hx, Hy, Hz at the centroid. The format of this file is the same between 2D and 3D designs.

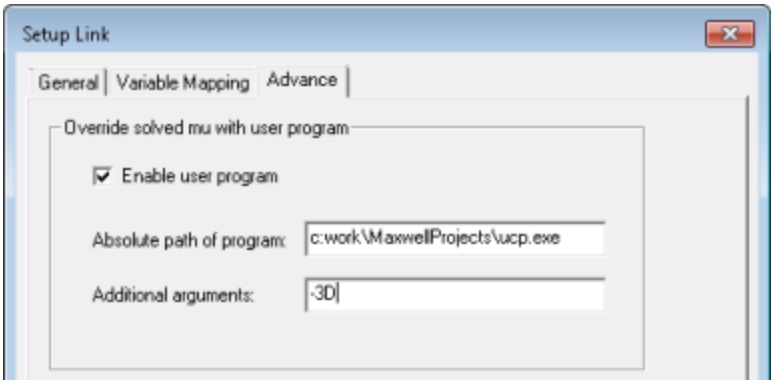
Note	Refer to Advanced Permeability Option File Formats for information on these file formats.

Considerations for using this feature

1. to set up the advanced user control program option for permeability, select **Setup Link** on the Solver tab to open the Setup Link dialog.



2. Specify the **Source Design** on the **General** tab.
3. On the **Advance** tab, select the check box **Enable user program** and provide the absolute path to the executable, and provide any **Additional arguments** needed by the user program.



4. Simulate the target design.

   Simulation of the source design will be automatically started if it is not already simulated. The user-supplied executable is launched. Additional information is logged in **Setup>Profile** regarding the launch of user executable.
5. Simulation of the target design continues after user executable exits.

Note the following:

- Field post processing of target design will use the user-overridden mu.
- Editing boundary/excitation setup in the target design invalidates the current solution in target design.
- Linked data remains valid when boundary/excitation is changed. On next simulation of the target design, the prior user-overridden permeability will be used.
- Edit of the link setup (this includes the path to the user control program, or “removing” the user control program will invalidate the current solution and also the cached linked data. On the next simulation of the target design, the “frozen” permeability will be extracted again from the source design. If the link setup has opted to enable the user control program, the currently designated user program will be launched. If the link setup has opted *not* to override the linked data, the extracted mu from the source design will be used as-is.
- **Clear Linked Data** in the target design invalidates the current solution and cached linked data in the target design.
- On the next simulation of the target design, the frozen permeability will be extracted again from the source design. The user executable will be launched to allow the permeability to be overridden.
- Editing the source design has no effect on the existing solution in the target design. Users need to **Clear Linked Data** in target design if a new simulation in the target design is desired.
- After the simulation of the target design, the user-overridden “mu” becomes part of the solution of the target design and this target design can then be used as the source of another design. This is explained in more detail with the following workflow.

Advanced Permeability Option File Formats

The following are the file formats for this feature:

3D Per mesh element permeability value.

This is the solved mu (i.e., the apparent permeability not the incremental permeability) from the 3D source design.

```
SolidID  6   14
h  230  79578.5
h  231  79578.5
h  232  79578.5
h  233  79578.5
h  234  79578.5
h  235  79578.5
h  236  79578.5
h  237  79578.5
h  238  79578.5
h  239  79578.5
h  240  79578.5
h  241  79578.5
h  242  79578.5
h  243  79578.5
E_DATA
```

The average permeability calculated by Maxwell solver. The average is calculated from the value at each of the nodes of the mesh element.

In the above example, SolidID, h, and E_DATA are keywords. “6” is the object ID. “14” is the number of elements in the object with ID “6”. “230”, “231”, etc. are element IDs.

Note that there could be multiple "blocks" of the above data if there are multiple non-linear objects in the design, as illustrated below.

```
SolidID  6   17
h  351  79578.5
h  352  79578.5
.
.
.
h  366  79578.5
h  367  79578.5
SolidID  93   12
h  1954  79578.5
h  1955  79578.5

.
.
.
h  1963  79578.5
h  1964  79578.5
h  1965  79578.5
E_DATA
```

2D Per mesh element permeability value.

This is the solved mu from the 2D source design.

```
The format is:
<number of total elements>
```

```
<objectID> <elementID> <X/R center position of element> <Y/Z center
position of element> <mu>
```

For example, the following solved mu indicates that there is ONE object that has TWO mesh elements. The first element has a mu value of 2.7482 and the second element has a mu of 2.76553.

```
2
76 1   0.000866667  6.66667e-005 2.7482
76 15  0.000733333 -6.66667e-005 2.76553
end
```

The following solved mu indicates a total of 16 mesh elements that belong to two objects.

```
16
64 24   0.000733333 -0.000533333 1.90406
64 27   0.000866667 -0.000266667 1.88459
64 28   0.000733333 -0.000266667 1.89191
64 41   0.000666667 -0.000333333 1.88789
64 42   0.000666667 -0.000466667 1.8909
64 53   0.000933333 -0.000333333 1.87607
64 69   0.000833333 -0.0005 1.91924
64 70   0.000966667 -0.0005 1.93687
64 71   0.0009 -0.000433333 1.89957
64 74   0.000933333 -0.000566667 2.0507
64 75   0.000866667 -0.000566667 1.95098
76 20   0.000666667 0 1.94237
76 21   0.000933333 0 1.92467
76 22   0.0008 0.000133333 2.06578
76 25   0.000866667 -0.000133333 1.89876
76 26   0.000733333 -0.000133333 1.90766
end
```

Per mesh element centroid location (x, y, z), Bx, Bx, By, Hx, Hy, Hz at the centroid. (All values in SI units.)

```
SolidID 6 14
h 230 x y z Bx By Bz Hx Hy Hz
.
.
.
E_DATA
```

Related Topics

"Using the Advanced Permeability Option " on page 16-12

Eddy Current Solver Settings

For an eddy current solution:

1. For 2D and 3D designs, enter a frequency value in the **Adaptive Frequency** text box, and select the desired units. Use of variables and expressions are also supported.
2. For 2D and 3D designs, enter a value in the **Nonlinear Residual** text box.
3. For 2D and 3D designs, you can select the **Smooth BH Curve** check box. This allows the solver to "smoothen" the list of points to make a refined BH curve for better solving. If this option is not selected, the solver just uses the discrete points.
4. For 3D designs, you can optionally Enable an Iterative Solver.
5. For 2D and 3D designs, you can select the **Number of Nonlinear Iterations** check box. This provides an option for you to set both a **Minimum** iteration number to avoid non-converged solution, and a **Maximum** iteration number to avoid taking too much computation time. These values must be integers greater than 0. By default the values for **Minimum** and **Maximum** are 1 and 100, respectively. Maximum number should be greater than the Minimum number.
6. For 3D designs, you can optionally choose to **Use higher order shape functions**. Enabling the higher order option gains better accuracy for eddy current regions.

   > **Note:** Higher order shape functions are not supported for a project with windings. For this case the solver falls back to lower order shape functions.

7. For 3D designs, you can optionally choose to **Use pre-computed permeability data**. Refer to Using the Permeability Link Option for Eddy Current Solutions for details on this option.
8. If desired, you can Import Mesh.

Related Topics

Using the Permeability Link Option for Eddy Current Solutions

Import Mesh

Enabling the Iterative Solver

Using the Permeability Link Option for 3D Eddy Current Solutions

For projects with materials whose permeability is nonlinear, setting the permeability link in a 3D Eddy Current design enables it to use the permeability data calculated in a source 3D magnetostatic design, whose matrix/permeability computation setting is **Apparent** (refer to Matrix/Permeability Tab Computation Settings for Magnetostatic Solutions). In this scenario, the nonlinear analysis is only performed in the 3D magnetostatic source design. In the 3D Eddy Current target design, the nonlinear material is treated as linear material using the “frozen” permeability data from the source design. The target design will be solved based on the mesh and permeability data imported from the source design. Small signal analysis is a typical application for this feature.

The permeability link option can also serve as the link if incremental permeability is desired in Eddy Current solutions by setting the matrix/permeability computation as **Incremental** in the source design settings (refer to Matrix/Permeability Computation Tab Settings for Magnetostatic Solutions).

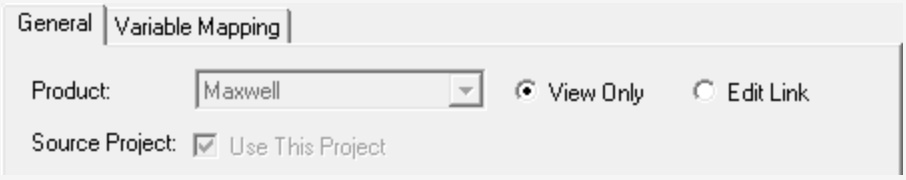
Prerequisites

- Source design - Maxwell 3D Magnetostatic
- Target design - Maxwell 3D Eddy Current
- The geometry of the source and target designs must be identical.

To set up and use the pre-computed permeability link for projects with nonlinear material:

1. Create a 3D Magnetostatic source design using nonlinear material.
2. Solve the 3D Magnetostatic design.
3. Create a 3D Eddy Current target design, whose geometry and materials are identical to the source magnetostatic design.
4. On the Eddy Current **Solve Setup** dialog box, **Solver** tab, select **Use pre-computed permeability data**, to use the permeability data that has been frozen from a previously solved 3D magnetostatic source design. Selecting this option enables the **Setup Link** button, and the **Setup Link** dialog opens, allowing you to specify the 3D magnetostatic source project, design, and solution containing the desired information. The mesh between the two linked designs is the same.

   By default, the **Setup Link** dialog opens on the **General** tab with only the **This project** radio button selected.

Note	If a link had previously been set up, the **Setup Link** dialog opens in **View Only** mode with all settings disabled.  To enable settings, you can select **Edit Link**.

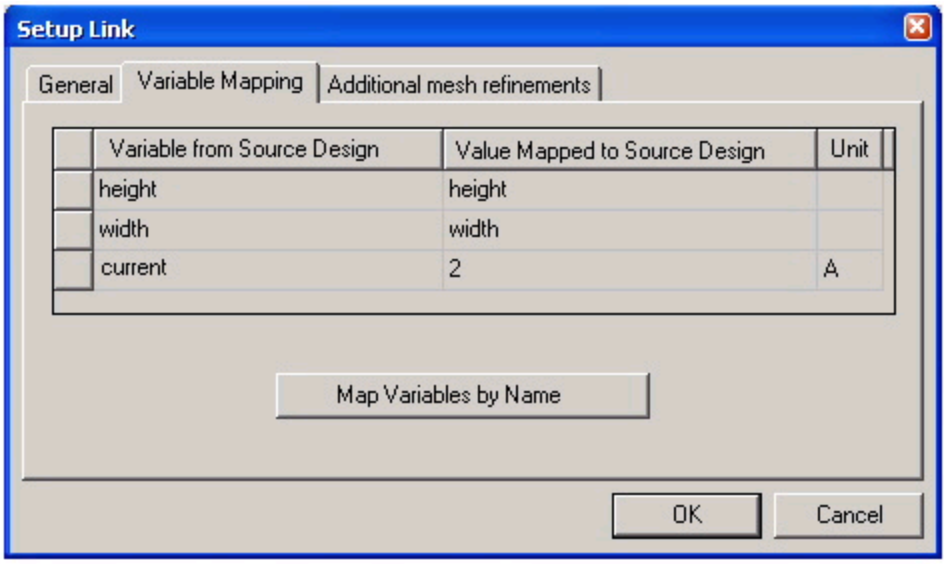
5. On the **Setup Link** dialog **General** tab, select the source 3D magnetostatic design and solution. The geometry of the source and target designs must be identical.
   - To use the current project as the source, check **Use This Project**. This disables the **Save source path relative to:** radio buttons and the ellipsis **[...]** button and its associated text field.
   - To specify a source project file other than the current project click the ellipsis **[...]** button to open a file browser window. When you have selected the project, click the **Open** button to accept the project file for the setup. You can use the check box to **Open as read only**.

Use the radio buttons to specify whether to save the source path relative to **The project directory of selected product** or **This project**.

6. Use the check box to specify whether to **Simulate source design as needed**.
7. Use the check box to specify whether to **Preserve source design solution**. Note that in the extractor mode, the source project will be saved upon exit. Extractor mode means that the software is opened during the link solely for the purpose of solving.
8. The **Variable Mapping** tab lets you view any variables contained in the selected Project.

   When there are variables in the source design, you can choose to "map" these variables to constant values, expressions, or variables in the target designs. Variable mapping becomes more important when the datalink type requires source and target design to be geometrically identical and the source design is geometrically parameterized.

   For linked designs with variables of the same name, you can click **Map Variables by Name** to automatically map same named variables. In this example the variables "height" and "weight" are mapped to the Source, whereas the "current" variable does not have a same named counterpart, and retains its value in the source design.



9. Click **OK** to accept the setup and close the Setup Link dialog and return to the Solve Setup dialog.
10. When you are finished specifying solution settings, click **OK** to close the **Solve Setup** dialog box and finalize the solution setup.
11. Solve the target 3D Eddy Current design.

Note	If the geometry of the source and target designs is not identical, an error message displays.

Related Topics

Eddy Current Solver Settings

Matrix/Permeability Tab Computation Settings for 3D Magnetostatic Solutions

Electrostatic Solver Settings

For an electrostatic solution you can:

- For 3D designs, you can Enable an Iterative Solver.
- Import Mesh.

AC Conduction Solver Settings

1. 1. For 2D and 3D designs, enter a frequency value in the **AdaptiveFrequency** text box, and select the desired units. Use of variables and expressions are also supported.
2. For 3D designs, enter a value in the **Nonlinear Residual** text box. This value specifies how close each solution must come to satisfying the equations that are used to compute the electric field.
3. For 3D designs, selecting the **Number of Nonlinear** Iterations check box allows you to set both a **Minimum** iteration number to avoid non-converged solution, and a **Maximum** iteration number to avoid taking too much computation time. These values must be integers greater than 1. The default values for **Minimum** and **Maximum** are 2 and 50, respectively. The **Maximum** number should be greater than the **Minimum** number.
4. If desired, you can Import Mesh.

DC Conduction Solver Settings

For a DC Conduction solution:

1. If desired, you can Import Mesh.
2. For 3D designs: optionally Enable an Iterative Solver.

Import Mesh for Non-Transient Solutions

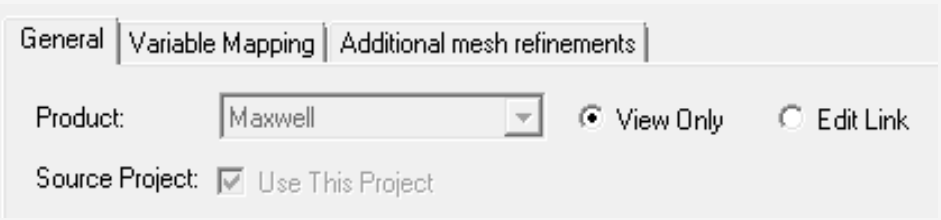
The **Import mesh** check box can be selected in the “target” design if the mesh for the same geometry is to be used in the analysis. This setup is designed to be generally applicable to all solutions.

If the **Import mesh** feature is selected, the **Setup Link** button becomes active, the **Setup Link** dialog box automatically appears, and the mesh will be imported from the source design to the target design.

Note	The usual way to create geometrically equivalent designs is by copying all geometry objects from one design and then using **Import From Clipboard** in the other design.

1. By default, the **Setup Link** dialog opens on the **General** tab with only the **This project** radio button selected.

Note	If a link had previously been set up, the **Setup Link** dialog opens in **View Only** mode with all settings disabled.



To enable settings, you can select **Edit Link**.

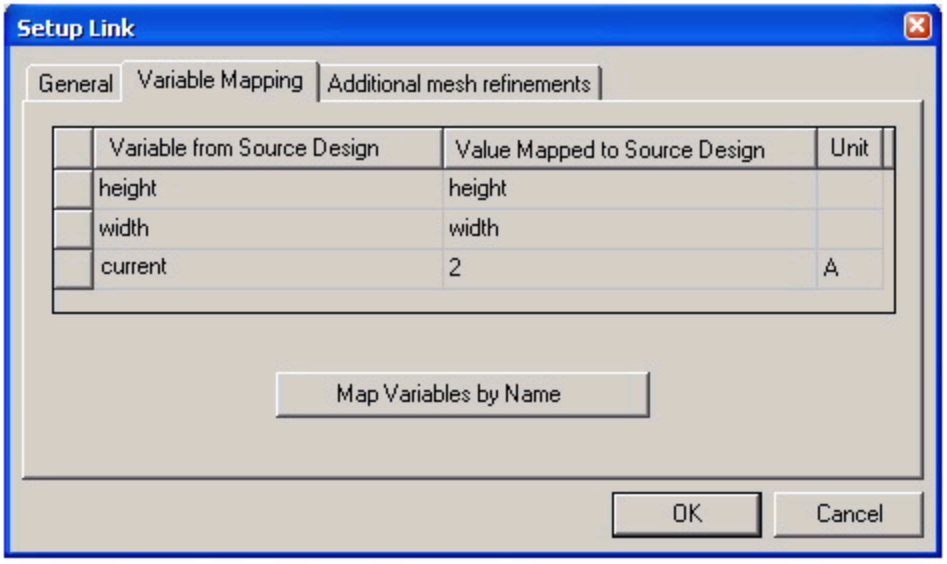
2. To select a source project do one of the following:
   - To use the current project as the source, check **Use This Project**. This disables the **Save source path to:** radio buttons and the ellipsis **[...]** button and its associated text field.
   - To specify a source project file other than the current project click the ellipsis **[...]** button to open a file browser window. When you have selected the project, click the **Open** button to accept the project file for the setup. You can use the check box to **Open as read only**.

     Use the radio buttons to specify whether to save the source path relative to **The project directory of the selected project** or **This project**.
3. When you select a source project file, the **Source Design** and the **Source Solution** fields are filled in with default values, and their drop down menus contain any available designs and solutions. You can use the drop down menus to choose from the available designs and solutions.

   The “Default” solution is the product-dependent solution of the first Setup. That is the setup listed first in the source design's project tree (alphanumerical order). A product-specific solution of this setup becomes the default solution. In most products, it is “LastAdaptive.” In a Transient solution type, it is “Transient.”
4. Use the check box to specify whether to **Simulate source design as needed**.
5. Use the check box to specify whether to **Preserve the source design solution**. Note that in the extractor mode, the source project will be saved upon exit. Extractor mode means that the software is opened during the link solely for the purpose of solving.
6. The **Variable Mapping** tab lets you view any variables contained in the selected Project.

   When there are variables in the source design, you can choose to “map” these variables to constant values, expressions or variables in the target designs. Variable mapping becomes more important when the datalink type requires source and target design to be geometrically identical and source design is geometrically parameterized.

   For linked designs with variables of the same name, you can click **Map Variables by Name** to automatically map same named variables. In this example the variables “height” and “weight” are mapped to the Source, whereas the “current” variable does not have a same named counterpart, and retains its value in the source design.



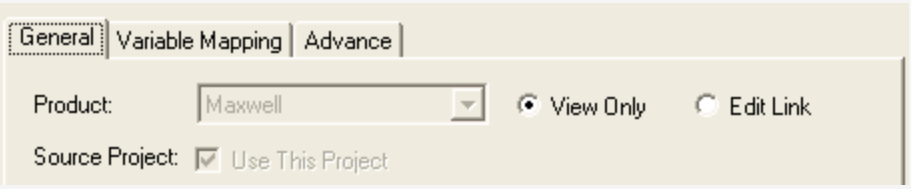
7. You can select the **Additional mesh refinements** tab to either **Apply mesh operation in target design on the imported mesh** (default), or **Ignore mesh operations in target design**.
8. Click **OK** to accept the setup and close the **Setup Link** dialog box.

Setup Link for Non-Transient Solutions

If you selected the **Use pre-computed permeability data** option (refer to *Permeability Options for Magnetostatic Solutions* or to *Using the Permeability Link Option for Eddy Current Solutions*), or **Import mesh**, do the following to specify the source project, design, and solution containing the information you want to use. You can also view and map variables and, for Import Mesh, specify additional mesh refinements.

General Tab

1. By default, the **Setup Link** dialog opens on the **General** tab with only the **This project** radio button selected.

Note	If a link had previously been set up, the **Setup Link** dialog opens in **View Only** mode with all settings disabled.  To enable settings, you can select **Edit Link**.

2. To select a source project do one of the following:

- To use the current project as the source, check **Use This Project**. This disables the **Save source path to:** radio buttons and the ellipsis **[...]** button and its associated text field.
- To specify a source project file other than the current project click the ellipsis **[...]** button to open a file browser window. When you have selected the project, click the **Open** button to accept the project file for the setup. You can use the check box to **Open as read only**.

Use the radio buttons to specify whether to save the source path relative to **The project directory of the selected project** or **This project**.

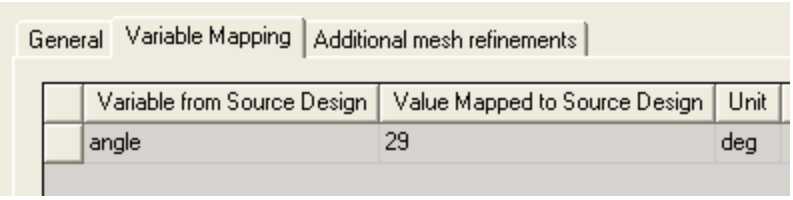
3. When you select a source project file, the **Source Design** and the **Source Solution** fields are filled in with default values, and their drop down menus contain any available designs and solutions. You can use the drop down menus to choose from the available designs and solutions.

   The “Default” solution is the product-dependent solution of the first Setup. That is the setup listed first in the source design's project tree (alphanumerical order). A product-specific solution of this setup becomes the default solution. In most products, it is “LastAdaptive.” In a Transient solution type, it is “Transient.”
4. Use the check box to specify whether to **Simulate source design as needed**.
5. Use the check box to specify whether to **Preserve the source design solution**. Note that in the extractor mode, the source project will be saved upon exit. Extractor mode means that the software is opened during the link solely for the purpose of solving.

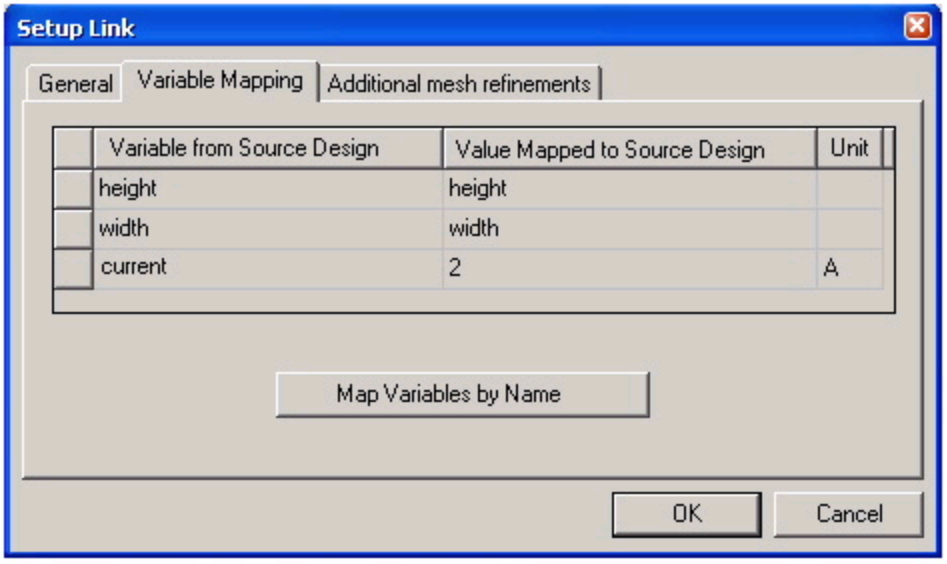
Variable Mapping Tab

The **Variable Mapping** tab lets you view any variables contained in the selected Project.

When there are variables in the source design, you can choose to “map” these variables to constant values, expressions or variables in the target designs. Variable mapping becomes more important when the datalink type requires source and target design to be geometrically identical and source design is geometrically parameterized.



For linked designs with variables of the same name, you can click **Map Variables by Name** to automatically map same named variables. In this example the variables “height” and “weight” are mapped to the Source, whereas the “current” variable does not have a same named counterpart, and retains its value in the source design.



Additional mesh refinements Tab

Note	This tab is present only for **Import mesh** link operations.

If you are setting up an Import mesh link you can use the radio buttons on the **Additional mesh refinements** tab to either **Apply mesh operation in target design on the imported mesh** (default), or **Ignore mesh operations in target design**.

Advance Tab

Note	This tab is present only when using "Permeability Options for Magnetostatic Solutions " on page 16-9 .

If you are setting up a link for the magnetostatic solver **Use pre-computed permeability data** option, you can select the **Advance** tab to override the solved mu value from the source file with a user-specified control program. Refer to "Permeability Options for Magnetostatic Solutions " on page 16-9 for details on using this feature:

Related Topics

"Permeability Options for Magnetostatic Solutions " on page 16-9

Using the Permeability Link Option for Eddy Current Solutions

"Import Mesh for Non-Transient Solutions " on page 16-20

Enabling the Iterative Solver

The iterative solver provides an alternative to the direct solver. When you select the **Enable Iterative Solver** option, Maxwell invokes the iterative solver. Maxwell automatically switches to

the direct solver if the iterative solver fails in convergence.

For more detail, see the technical notes for *Iterative Matrix Solver*.

To enable the iterative solver:

1. On the **Solution Setup** dialog box **Solver** tab, check the **Enable Iterative Solve** check box.

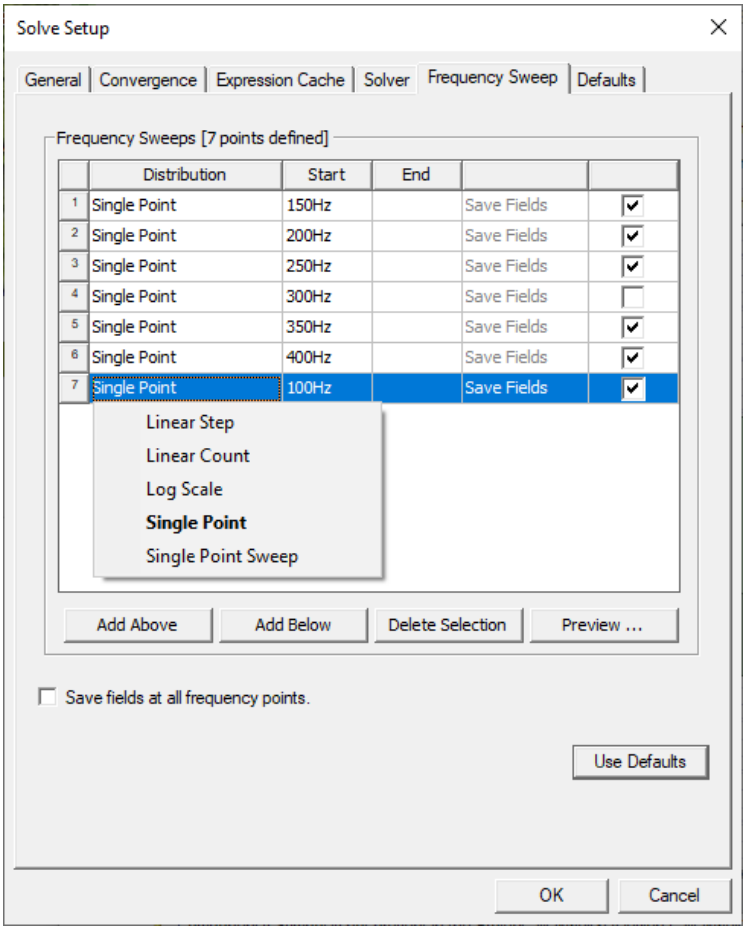
   This enables the **Relative Residual** input field.
2. Enter a value for the **Relative Residual**. The residual measures the convergence of the iterative solver. The default value is **1e-6** for the magnetostatic and electrostatic solvers and **1e-4** for the eddy current solver.

Related Topics

Technical Notes: *Iterative Matrix Solver Technical Details*

Defining Settings on the Frequency Sweep Tab for Eddy Current and AC Conduction Solutions

On the **Frequency Sweep** tab of the **Solve Setup** dialog box:



1. Use the **Add Above**, **Add Below**, and **Delete Selection** buttons to add or remove sweeps. Each row represents a separate sweep. Selecting a sweep's **Distribution** activates a drop-down menu allowing you to select the type of sweep. Information in the columns varies with the type of sweep.
    - **Linear Step** – A linear range of frequency points for which you specify a constant step size. Maxwell will solve the frequency point at each step in the specified frequency range, including the start and stop frequencies. For example, specifying 10Hz for the **Start** frequency, 20Hz for the **End** frequency, and 2.5Hz for the step size instructs Maxwell to compute a solution for frequencies of 10, 12.5, 15, 17.5, and 20 Hz.
    - **Linear Count** – A linear range of frequency points for which you specify the number (count) of points within the frequency range defined by the **Start** and **End** frequencies. Maxwell will divide the frequency range, which includes the start and end frequencies, by the count you specify and solves each frequency point in the count.
    - **Log Scale** – A range of points for which you specify the **Start**, **End**, and number of samples. Maxwell assigns the sampled points using intervals based on a logarithmic

scale.

- **Single Point** – Individual frequency points for which you specify the **Start** value only. A check mark in the **Save Fields** column indicates that the fields for the point will be saved.
- **Single Point Sweep** – Converts the current selection to single points. This is convenient if you want to save (or not save) fields at specific frequency points.

2. Click the **Preview** button to display the currently defined sweep(s).
3. Optionally, you can check **Save fields at all frequency points** to save fields for the entire set of sweep ranges – except for **Single Point** entries that are unselected.

Defining Settings on the Defaults Tab for Non-Transient Solutions

Default analysis parameters are used to initialize the data for new setups.

To set up the default analysis parameters, define the following parameters under the **Defaults** tab of the **Solve Setup** dialog box:

1. Click **Save Defaults** if you want to save the defaults based on the data for the current setup.
2. Click **Revert to Standard Defaults** to clear the existing values.
3. Click **OK**.

Setting Analysis Parameters for Transient Solutions

When you set up an analysis, define the parameters on the following tabs of the **Solve Setup** dialog box (which appears when you click **Maxwell**>**Analysis Setup>Add Solution Setup**):

- General
- Save Fields
- Advanced
- Solver
- Expression Cache
- Defaults

Defining Settings on the General Tab for Transient Solutions

The following settings can be defined on the **General** tab of the **Solve Setup** dialog box for transient solutions:

- Setup Name
- Transient Setup (2D) or Transient Setup (3D)
- HPC and Analysis Options

Setup Name for Transient Solutions

By default, **Setup1** appears as the name of the first analysis you set up.

To change the name of the setup, type the new name in the **Name** text box on the **General** tab of the **Solve Setup** dialog box.

To disable the setup, uncheck the **Enabled** check box. Refer to *Disabling and Enabling an Analysis Setup* for additional information.

Transient Setup (2D)

To set up a 2D transient analysis, define the following parameters on the **General** tab of the **Solve Setup** dialog:

Adaptive Time Step

1. Determine whether you want the software to adaptively alter the time step during the simulation to optimize the solution for speed and accuracy. If so, check the **Adaptive Time Step** check box. If not, continue with step _ .
2. Enter the **Stop Time**, which is the value at which the solution will stop, and select the units.
3. Enter the **Initial Time Step**. This will be the time step used to start the simulation and begin the adaptive process.
4. Enter the **Maximum Time Step** and **Minimum Time Step** to set the allowable range the adaptive process use to alter the time step
5. Enter a value for the **Error Tolerance**. Generally this value should be smaller than the minimum time step specified by at least an order of magnitude.

Non-Adaptive Time Step

1. Enter the **Stop Time**, which is the value at which the solution will stop, and select the units.
2. Enter the **Time Step** for the analysis, which defines the time increments, and select the units. The time step can also be defined as the function of time, speed, and position.

Note	When you have an external circuit, the Time Step is controlled from the circuit simulator, and the time step settings in the solution setup dialog are ignored. This can be controlled using the **Maxwell 2D>Excitations>External Circuit>Set Minimum Time Step command**. Time Step settings in the solution setup dialog box are also ignored for transient-transient simulation in which case Twin Builder controls the time step.

Transient Setup (3D)

To set up a 3D transient analysis, define the following parameters on the **General** tab of the **Solve Setup** dialog box:

1. Enter the **Stop Time**, which is the value at which the solution will stop, and select the units.
2. Enter the **Time Step** for the analysis, which defines the time increments, and select the units. The time step can also be defined as the function of time, speed, and position.

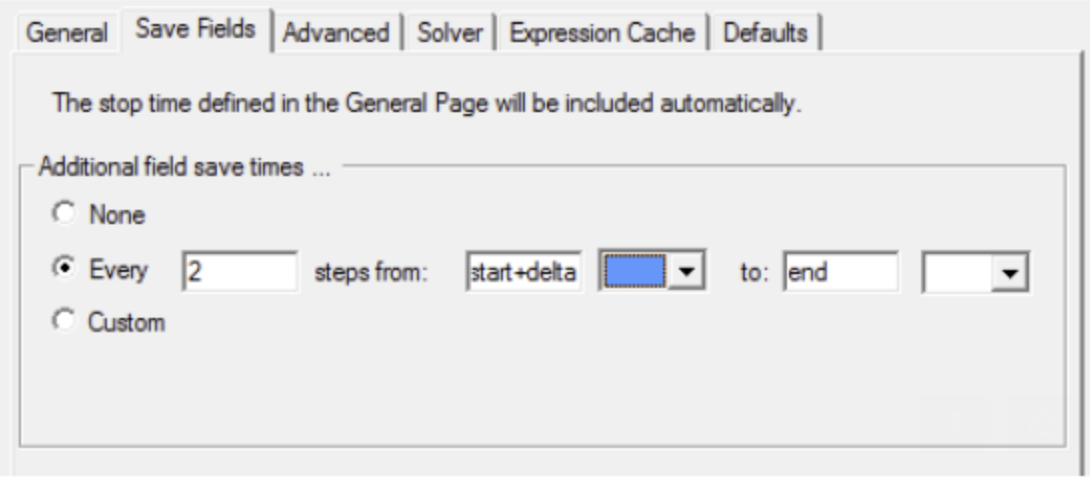
Note	When you have an external circuit, the Time Step is controlled from the circuit simulator, and the time step settings in the solution setup dialog are ignored. This can be controlled using the **Maxwell 3D>Excitations>External Circuit>Set Minimum Time Step command**. Time Step settings in the solution setup dialog box are also ignored for transient-transient simulation in which case Twin Builder controls the time step.

Defining Settings on the Save Fields Tab for Transient Solutions

To specify the time points at which to save fields, do the following on the **Save Fields** tab of the **Solve Setup** dialog box:

1. Specify the field save times:
   - **None** - only the stop time is saved. This is the default.
   - Every **N** time steps - Enter the number of steps, start time, stop time and select the units. The values can be constants, variables or expressions.

     For example, if the variable start is 2ms, delta is 1ms, and end is 12ms, and the solution is calculated every ms, we would save fields at 3ms, 5ms, 7ms, 9ms, 11ms, and 12ms.



   - **Custom** - select this option to add time points as desired.
2. For **Custom**, click the **Preview** button to view the save field times. Some cases do not support a preview (for example, when variables are used to define the settings or when a user control program is used). You can add and remove rows at your desired locations using the **Add Above**, or **Add Below**, and **Delete Selection** buttons.
3. Click **OK**.

Defining Settings on the Advanced Tab for Transient Solutions

The following settings can be defined on the **Advanced** tab of the **Solve Setup** dialog box:

- Control Program
- Start/Continue from a previously solved setup
- Import mesh

Related Topics

Nonlinear Materials

"Setup Link for Transient Solutions " on page 16-42

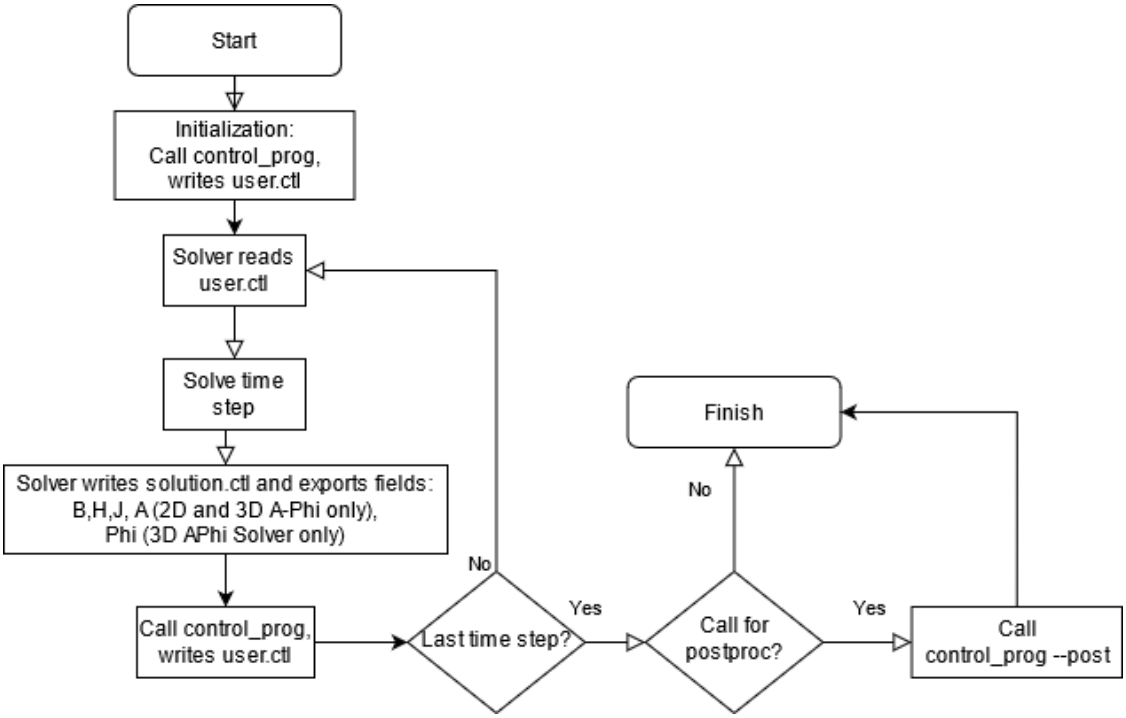
Using a 2D Eddy Current Solution as the Initial Condition for a Transient Solution

Using a Control Program in Maxwell 2D and 3D Transient Solutions

User control programs are externally created executable files, or Python (`*.py`) scripts that are called after each time step, and allow you to control the source input, circuit elements, mechanical quantities, time step, and stopping criteria, based on the updated solutions.

> **Note:** The control program is executed from a temp directory that Maxwell creates for every setup run (For parametric setup a temp folder is create for each task). The solver copies a control program file with name **<setup_name>.py** for Python, or **<setup_name>.ctrlprog** for compiled program to the temp directory.

The user control program uses the file with fixed name **user.ctl** to output control parameters to control the execution of the transient solver. Each file must use a predefined syntax which is flexible enough to cover a wide range of items.



The process shown in the flowchart can be summarized as:

1. The solver calls the user-control program on initialization.
2. The user-control program writes control information to file **user.ctl**.
3. The transient solver reads control parameters from **user.ctl**.
4. The transient solver solves the current time step.
5. The transient solver copies the previous solution to **previous.ctl** and writes out solution information to **solution.ctl**. If requested, the solver exports:
   - B, H, J fields (**bfield.ctl**, **hfield.ctl**, **jfield.ctl**, respectively)
   - A field **afield.ctl** (available in 2D and 3D A-Phi only)
   - Phi field **phifield.ctl** (available in 3D A-Phi only).
6. Return to the transient solver if the control program succeeds with exit status 0 or fails with exit status non-zero.
7. Return to step 1 for the next time step.
8. (Optional) Call script after last time step for post processing. If user selects the **Call after last time step for post processing** check box (refer to *Activating a Control Program in Transient Solutions*), then the control program is called after the simulation is completed.

   If you use a Python control program, the script is called with `--post` flag

   (for example: `my_script.py --post`).

   For a compiled control programs the script is called with `-post` flag

   (for example: `my_tool.exe -post`).

Note: By default a control program script will be called by the pre-installed Python interpreter:
`<install_path>\Win64\commonfiles\CPython\37\winx64\Release\python\python.exe`
however, the user can specify a custom python interpreter to be used by setting following environment variable:
EM_CTRL_PROG_PYTHON_PATH=<*path_to*\python.exe>

Related Topics

"Activating a Control Program in Transient Solutions " below
"Control Program File Formats in Transient Solutions " on the facing page

Activating a Control Program in Transient Solutions

Use the following steps to access and invoke a user-control program to use with the Maxwell 2D or 3D transient solver. User control programs can be either externally created executable (`.exe`) files, or Python (`*.py`) scripts.

To specify a control program to use to generate the solution:

1. Select the **Use Control Program** check box.

   The control program field becomes active, allowing you to enter the name of the user control program.
2. Click the **...** button to select the user control program you want to use during the solution generation process.

   The **Open** window appears.
3. Select the control program, and click **Open** to return to the **Advanced** tab of the **Solve Setup** dialog box.
4. Enter the arguments to pass to the control program in the **Arguments** text box. The solver calls the program in the following format:

   ```
   program_name specified_arguments
   ```
5. If you select the **Call after last time step for post processing** check box, then the control program is called after the simulation is completed.

   If you use a Python control program, the script is called with `--post` flag

   (for example: `my_script.py --post`).

   For a compiled control programs the script is called with `-post` flag

   (for example: `my_tool.exe -post`).
6. Click **OK** to accept the configuration or **Cancel** to ignore the settings. You return to the **Solve Setup** window.

Related Topics

"Control Program File Formats in Transient Solutions " below

"Using a Control Program in Maxwell 2D and 3D Transient Solutions " on page 16-30

Control Program File Formats in Transient Solutions

User control programs and related files have their own file formats, which should be followed when creating control program files.

- User Control File Format
- Solution and Previous Control File Formats
- User Control Fields Extraction Format
- Output variable values are output to a file with a fixed name of **outputvar.ctl**. The file contents use the following format:

```
<output variable name> <output variable value>
<output variable name> <output variable value>
...
```

The following summarizes the files related to user control programs for transient solutions.

File Name	Functionality
user.ctl	Input data that controls Maxwell Solver
solution.ctl	Data for global quantities such torque, current, speed etc
previous.ctl	Same data as solution.ctl but for previous time step
outputvar.ctl	Contains expression cache variables
var.data	Variations information from design and project variables
afield.ctl	Magnetic potential (available only in 2D and 3D A-Phi solver)
bfield.ctl	Magnetic flux density
hfield.ctl	Magnetic field strength
jfield.ctl	Electric current density
phifield.ctl	Electric Scalar Potential (available only in 3D A-Phi solver)

User Control File Format

The **user.ctl** file is created by the user control program. Below are descriptions of the keywords that can be used in a **user.ctl** file, and indicating the solver types for which they are valid.

Keyword	Description	2D	3D	3D A-Phi
windingSrc <winding_name> <val>***	Voltage [V]/Current [A] value for Winding	X	X	X
windingR <winding_name> <val>***	Resistance value [Ohm] for Voltage Winding	X	X	X
windingL <winding_name> <val>***	Inductance value [H] for Voltage Winding	X	X	X
loadTorque <value>	Load Torque [N*m] for rotational motion only	X	X	
loadForce <value>	Load force in MotionSetup [N] for translational only	X	X	
damping <value>	Damping coefficient [translational: N*s/m; rotation: N*m*s/rad]	X	X	
loadInertia <value>	Load Inertia in MotionSetup [kg*m²] for rotational only	X	X	
mass <value>	Mass in MotionSetup [kg] for translational only	X	X	
speed <value>	Speed of the moving object [translation - m/s; rotation - rad/s]	X	X	
position <value>	Position of the moving object [[translation - user unit (from Motion Setup); rotation – rad]	X	X	
timeStep <value>	Time step size [s]	X	X	X
stop <0 or 1>	Stop solver if value is equal to 1	X	X	X
exportFieldAtMeshNodeOnAllObjects	Request to export fields on all object in design	X	X	X
exportFieldAtMeshNodeOnObject <objectID>*	Request to export fields on specific object	X	X	X
NStep <value>	Export fields only every N steps (works only in conjunction with exportFieldAtMeshNode<...>)	X		
FieldSelect <value>**	Export only specific field (one of A,B,H,J) (works only in conjunction with exportFieldAtMeshNode<...>)	X		

Keyword	Description	2D	3D	3D A-Phi
Binary <0 or 1>	Export fields in binary format (works only in conjunction with exportFieldAtMeshNode<...>)	X		
voltageSrc <voltage_name> <source_value>***	Voltage source value [V]			X
currentSrc <current_name> <source_value>***	Current source value [A]			X
Full line could be repeated for *multiple object IDs; **multiple fields; ***multiple excitation names				

The following **user.ctl** file example:

- Sets the time step to 5ms.
- Sets Resistance to 1µOhm and Inductance to 1µH for Winding A and Winding B.
- Sets Voltage for Winding A to 10 V and for Winding B to -10 V.
- After time step is completed all fields for the object with ID 88 will be exported.

```
begin_data
 windingSrc WindingA 10
windingR WindingA 1e-6
windingL WindingA 1e-6
windingSrc WindingB -10
windingR WindingB 1e-6
windingL WindingB 1e-6
exportFieldAtMeshNodeOnObject 88
timeStep 5e-3
stop 0
end_data
```

Related Topics

"Using a Control Program in Maxwell 2D and 3D Transient Solutions " on page 16-30
"Activating a Control Program in Transient Solutions " on page 16-32

User Control Fields Extraction Format

The **user.ctl** file is created by the user control program. Depending on the solver the following field types are available during export:

Field	2D	3D	3D A-Phi
B – Magnetic Flux Density	X	X	X
H – Magnetic Field Strength	X	X	X
J – Current Density	X	X	X
A – Magnetic Vector Potential	X		X
Phi - Electric Scalar Potential			X

To extract fields from Maxwell, the control program must create a **user.ctl** file with one of the two following syntax strings:

> **Note:** You can combine field export keywords with other keywords.

Extract fields on all objects

user.ctl

```
begin_data
exportFieldAtMeshNodeOnAllObjects
end_data
```

Extract fields by object ID

user.ctl

```
begin_data
exportFieldAtMeshNodeOnObject <objectID>
exportFieldAtMeshNodeOnObject <objectID>
end_data
```

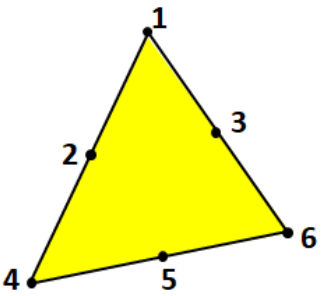
You can get the desired object IDs as follows:

1. Select **Tools -> Open Command Window**.
2. In the following example code snippet, replace "Stator" in the `object_name` with the name of the desired object.

```
object_name = "Stator"
oProject = oDesktop.GetActiveProject()
oDesign = oProject.GetActiveDesign()
oEditor = oDesign.SetActiveEditor("3D Modeler")
object_id = oEditor.GetObjectIDByName(object_name)
print(object_id)
AddInfoMessage(str(object_id))
```

3. Copy and paste the code snippet into the Command Window and execute.
4. Repeat as needed for additional objects.

Data Structure and Nodes Order for 2D Fields Extraction



Node order in Maxwell 2D element

The Maxwell 2D solver writes the data and updates field files (**afield.ctl**, **bfield.ctl**, **hfield.ctl**, **jfield.ctl**) at each time step with the following structure:

Time
M
ID K x_1 y_1 x_4 y_4 x_6 y_6 B_{x1} B_{y1} B_{x2} B_{y2} B_{x3} B_{y3} B_{x4} B_{y4} B_{x5} B_{y5} B_{x6} B_{y6}

where:

- Time – is the time corresponding to the time-stepping FE solution
- M – is the total number of elements
- ID – is the object ID
- K – is the element number
- x_i, y_i – are the coordinates of node i, for i in [1, 4, 6]
- B_{xi} B_{yi} – is the field X, Y component value of node i, for i in range 1 to 6

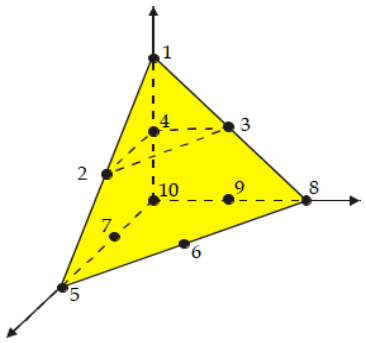
> **Note:** For vector potential, the **afield.ctl** file has one component of the field, which is A_z.

> **Note:** Coordinates of the element are given for the vertices (1, 4, 6) only. Middle points (2, 3, 5) can be calculated from the other given coordinates since they are located in the middle as shown above, for example: $x_2 = (x_1+x_4)/2$.

The following is an example for **bfield.ctl** data extracted in an object with ID 761 that consists of 2 elements at t = 0.04 sec.

```
0.040000
2
761 550 0.002 0.037 0.003 0.037 0.003 0.037 -2.220 0.488 -1.540 0.783 -1.799 0.950 -0.860 1.077 -1.119 1.245 -1.379 1.413
761 573 0.018 0.032 0.021 0.032 0.020 0.031 -0.695 1.266 -0.626 1.433 -0.810 1.378 -0.556 1.599 -0.740 1.544 -0.924 1.490
```

Data Structure and Nodes Order for 3D Fields Extraction



Node order in Maxwell 3D element

The Maxwell 3D solver writes the data and updates field files (**bfield.ctl**, **hfield.ctl**, **jfield.ctl**) at each time step with the following structure:

Time
M
ID K x_1 y_1 z_1 x_5 y_5 z_5 x_8 y_8 z_8 x_{10} y_{10} z_{10} B_{x1} B_{y1} B_{z1} B_{x2} B_{y2} B_{z2} … B_{xi} B_{yi} B_{zi} … B_{x10} B_{y10} B_{z10}

where:

- Time – is the time corresponding to the time-stepping FE solution
- M – is the total number of elements
- ID – is the object ID
- K – is the element number
- x_i, y_i, and z_i– are the coordinates of node i, for i in [1, 5, 8, 10]
- B_{xi} B_{yi}, B_{zi} – is the field X, Y, Z component value of node i, for i in range 1 to 10

> **Note:** The coordinates of the 3D element are given for the vertices (1, 5, 8, 10) only. Middle points (2, 3, 4, 6, 7, 9) can be calculated from the other given coordinates since they are located in the middle as shown above, for example: $x_2= (x_1+x_5)/2$

Related Topics

"User Control File Format " on page 16-34

"Using a Control Program in Maxwell 2D and 3D Transient Solutions " on page 16-30
"Activating a Control Program in Transient Solutions " on page 16-32

Solution and Previous Control File Formats

The **solution.ctl** and **previous.ctl** files are created by the transient solver. Below is a list of keywords and definitions applicable to these files.

Keyword	Description	2D	3D	3D A-Phi
time <value>	Current time in simulation [s]	X	X	X
position <value>	Position when motion setup is assigned [translation - user unit (from Motion Setup); rotation – rad]	X	X	
speed <value>	Speed when motion setup is assigned [translation - m/sec; rotation - rad/sec]	X	X	
torque moving_group <value>	Torque [N*m]	X	X	
force moving_group <value>	Force on moving objects [N]	X	X	
strandLoss <value>	Stranded loss for all objects [W]	X	X	X
solidLoss <value>	Solid loss for all objects [W]	X	X	X
coreLoss <value>	Core loss for all objects [W]	X	X	X
timeStep <value>	Time step size [s]	X	X	X
windingFlx <winding_name> <value>*	Winding Flux value [Wb]	X	X	X
windingEmf <winding_name> <value>*	Winding EMF value [V]	X	X	X
windingI <winding_name> <value>*	Winding Current value [A]	X	X	X
windingV <winding_name> <value>*	Winding Voltage value [A] (voltage windings only)	X	X	X
voltageV <voltage_name> <value>*	Voltage value on voltage excitation [V]			X
voltageI <voltage_name> <value>*	Current value on voltage excitation [A]			X
currentV <current_name> <value>*	Voltage value on current excitation [V]			X
currentI <current_name> <value>*	Current value on current excitation [A]			X
* Could be repeated for multiple excitation names				

An example `solution.ctl` file for 2D/3D solvers:

```
begin_data
 time <current_time>
 strandLoss <value>
 solidLoss <value>
 coreLoss <value>
 torque <torque>
 speed <speed>
 position <position>
 windingI <winding_name> <value>
 windingEMF <winding_name> <value>
```

```
 windingFlx <winding_name> <value>
end_data
```

An example `solution.ctl` file for the A-Phi solver:

```
begin_data
 time <current_time>
 strandLoss <value>
 solidLoss <value>
 coreLoss <value>
 windingI <winding_name> <value>
 windingEMF <winding_name> <value>
 windingFlx <winding_name> <value>
 voltageV <voltage_name> <value>
 voltageI <voltage_name> <value>
end_data
```

Related Topics

"Using a Control Program in Maxwell 2D and 3D Transient Solutions " on page 16-30

"Activating a Control Program in Transient Solutions " on page 16-32

Start/Continue from a Previously Solved Setup

Enabling the **Start/Continue from a previously solved setup** option allows the source design solution to be linked to a set of different target designs, where the user can continue the solution based on a different parameter/circuit setup. For example, to speed up the process of either a 2D or 3D transient solution reaching a steady state, you can use this option to link a 2D (or 3D) eddy current solution as the initial condition for a corresponding transient solution.

If the **Start/Continue from a previously solved setup** check box is selected, the **Setup Link** button becomes active, and the **Setup Link** dialog box automatically appears in which the user can link the current design to a project solution set based on a different parameter/circuit setup. Refer to "Setup Link for Transient Solutions " on the next page for details on setting up this link.

Note	**Power loss** and **Force density distribution** cannot be computed when **Start/Continue from a previously solved setup** is enabled.

Related Topics

Using an Eddy Current Solution as the Initial Condition for a Transient Solution

Setup Link for Transient Solutions

Import Mesh for Transient Solutions

The **Import mesh** check box can be selected in the "target" design if the mesh for the same geometry is to be used in the analysis. This setup is designed to be generally applicable to all

transient solutions. This feature is not necessarily related to the dynamic demagnetization distribution capability.

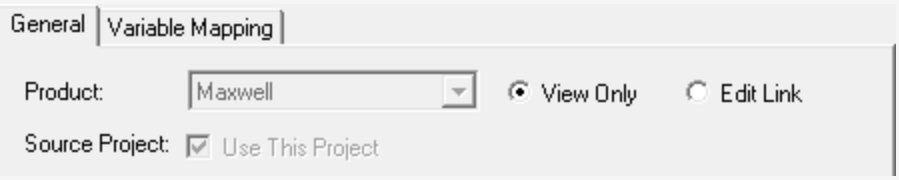
If the **Import mesh** feature is selected, the **Setup Link** button becomes active, the **Setup Link** dialog box, which includes an **Additional mesh refinements** tab automatically appears, and the mesh will be imported from the source design to the target design. Refer to "Setup Link for Transient Solutions " below for details on setting up this link.

Note	The usual way to create geometrically equivalent designs is by copying all geometry objects from one design and then using **Import From Clipboard** in the other design.

Setup Link for Transient Solutions

If you selected **Start/Continue from a previously solved setup**, or **Import mesh**, do the following to specify the source project, design, and solution containing the information you want to use. You can also view and map variables and, for Import Mesh, specify additional mesh refinements.

1. If the **Setup Link** dialog is not already open, click **Setup Link** (located at the bottom of the **Solve Setup** dialog box **Advanced** tab) to open the **Setup Link** dialog box.
2. By default, the **Setup Link** dialog opens on the **General** tab with only the **This project** radio button selected.

Note	If a link had previously been set up, the **Setup Link** dialog opens in **View Only** mode with all settings disabled.  To enable settings, you can select **Edit Link**.

3. To select a source project do one of the following:
   - To use the current project as the source, check **Use This Project**. This disables the **Save source path relative to:** radio buttons and the ellipsis **[...]** button and its associated text field.
   - To specify a source project file other than the current project click the ellipsis **[...]** button to open a file browser window. When you have selected the project, click the **Open** button to accept the project file for the setup. You can use the check box to **Open as read only**.

     Use the radio buttons to specify whether to save the source path relative to **The project directory of selected product** or **This project**.
4. When you select a source project file, the **Source Design** and the **Source Solution** fields are filled in with default values, and their drop down menus contain any available designs

and solutions. You can use the drop down menus to choose from the available designs and solutions.

The “Default” solution is the product-dependent solution of the first Setup. That is the setup listed first in the source design's project tree (alphanumerical order). A product-specific solution of this setup becomes the default solution. In most products, it is “LastAdaptive.”

5. Use the check box to specify whether to **Simulate source design as needed**.
6. Use the check box to specify whether to **Preserve source design solution**. Note that in the extractor mode, the source project will be saved upon exit. Extractor mode means that the software is opened during the link solely for the purpose of solving.
7. The **Variable Mapping** tab lets you view any variables contained in the selected Project.

   When there are variables in the source design, you can choose to “map” these variables to constant values, expressions or variables in the target designs. Variable mapping becomes more important when the datalink type requires source and target design to be geometrically identical and source design is geometrically parameterized.

   For linked designs with variables of the same name, you can click **Map Variable by Name** to automatically map same named variables.
8. The **Additional mesh refinements** tab, which is present only if you chose Import Mesh on the **Solve Setup** dialog **Solver** tab, allows you to choose either to **Apply mesh operations in target design on the imported mesh** (default) or **Ignore mesh operations in target design**.
9. Click **OK** to close the **Setup Link** dialog box and return to the **Solve Setup** dialog box **Advanced** tab.

Related Topics

Using a 2D Eddy Current Solution as the Initial Condition for a Transient Solution

"Start/Continue from a Previously Solved Setup " on page 16-41

"Import Mesh for Transient Solutions " on page 16-41

Using an Eddy Current Solution as the Initial Condition for a Transient Solution

In order to speed up the process of either a 2D or 3D transient solution reaching a steady state, you can set the solution from a 2D or 3D eddy current solver as the initial condition for the corresponding transient solver.

Note	Since the eddy current solution corresponds to a stationary operating condition (zero speed), in order to generate the initial condition for the transient solution with non-zero speed, a transformation of conductivity on the moving objects where eddy current is taken into account is required in the eddy current simulation. The transformation is:

$$\sigma_0 = \sigma \cdot s_n$$

where σ_0 is the transformed conductivity from speed n to speed zero, which is to be set in the eddy current simulation, σ is the real conductivity in transient simulation, and s_n is the slip, defined as:

$$s_n = \frac{n_s - n}{n_s}$$

where n_s is the synchronous speed.

If there is an end connection, the end resistance R used in the transient simulation also has to be referred to the value R_0 associated with zero speed before being used in the eddy current solution by:

$$R_0 = R / s_n$$

Prerequisites

- Source design - Maxwell 2D (or 3D) Eddy Current (Last Adaptive solution only)
- Target design - Maxwell 2D (or 3D) Transient
- The source and target designs must be identical (identical geometries, boundaries, excitations, windings, etc).

To setup and use a 2D (or 3D) eddy current solution as the initial condition for a corresponding 2D (or 3D) transient solution:

1. Create an eddy current source design, and ensure that **Solve Matrix: After last pass** is selected on the **General** tab of the eddy current **Solve Setup** dialog.
2. Solve the eddy current design.
3. Create a transient target design, whose geometry, boundary conditions, excitations, windings, etc., are identical to the the source eddy current design.
4. On the transient target design's **Solve Setup** dialog box, **Advanced** tab, click **Start/Continue from a previously solved setup** to open the **Setup Link** dialog.
5. On the **Setup Link** dialog **General** tab, select the source eddy current design and solution (Last Adaptive solution only). Refer to "Setup Link for Transient Solutions " on page 16-42 for details on setting up this link.
6. Solve the target transient design.

Note	• If the source and target designs are not identical, an error message displays. • If the eddy current source design has not been solved, an error message

	displays. • If a legacy eddy current design solved with a previous release of Maxwell is used as the source design, you must re-solve the eddy current design before solving the target transient design or an error message will be displayed.

Related Topics

Start/Continue from a Previously Solved Setup

Setup Link for Transient Solutions

Defining Expressions for Transient Solutions

You can specify additional evaluation criteria through the use of expressions and output variables.

To set expressions as evaluation criteria:

1. Right-click on the Analysis icon in the Project tree to open the **Solve Setup** dialog box.
2. Click the **Expression Cache** tab and click the **Add...** button to open the **Add to Expression Cache** dialog box.
3. Specify the Context for the expression, including **Report Type**, **Solution**, and **Parameter**.
4. Under the **Trace** tab, select from the **Category**, **Quantity**, and **Function** lists to create expressions.

   Selecting a listed category lists the Quantities and Functions available for each category. If you have defined one or more output variables, you can see them by selecting the **Output Variables** category. The **Output Variables** button opens a dialog box that lets you define output variables. See Specifying Output Variables.

   When you have created an expression, it appears in the **Expression** field of the **Trace** tab. If desired, you can use the **Range Function** button to select range functions to apply to the expression.

   Under the **Calculation Range** tab, you can view the values of available sweep variables. Clicking the ellipsis **[...]** button in the **Edit** column opens a list of values from which you can select.
5. When finished defining the expression that you want to add to the cache, click the **Add Calculation** button.

   This adds the selected expression and the associated context to a table in the **Expression Cache** tab. You can define additional expressions with contexts and add them in the same way.
6. When finished adding expressions, click **Done** to close the **Add to Expression Cache** dialog box.

   The **Expression Cache** tab lists the expressions you have added in a table.
   - The **Title** field is editable, by default showing the name as built from the expression (with underscores removed).

- The **Expression** field shows the full expression. If necessary, you can resize the **Solve Setup** dialog box. You can also resize each columns in the table.
- The **Context** column for non-field types lists the parameter type specifying the force, torque or matrix setup. For these types the context also can be given as part of the expression. For example, the expression *Force1.Force_x* with a context of *None* is equivalent to an expression of *Force_x* with a context of *Force1*.

For field quantities the **Context** can be the geometry on which to evaluate the field quantity - a point or line. If you use a line you need to use some range function to reduce the dimensionality to a single value (or use the Calculation Range tab to specify a single point along the line). Note that, if you create a named expression in the field calculator that already evaluates to a single value, then the **Context** here will likely be *None*.

- The **Intrinsics** column shows a clickable button that opens an **Edit Calculation Range** dialog box. If the **Intrinsics** column button shows **None** you cannot edit the value. If the button shows variables, click the ellipsis **[...]** in the **Edit** column to display a list of the variable values from which you can select. Click **OK** to close the **Edit Calculation Range** dialog box and apply your selections to the **Expression Cache**.

The **Intrinsics** field lists the values you may have set in the **Calculation Range** tab when you added the expression. This is used to set the values of “other” sweeps. Usually there are no other sweeps because the values of the variables are set in the definition of each variation. In some cases you may need to specify the values of “other” sweeps. For field values with a line for the context, you can specify the (normalized) distance along the line For eddy current field quantities you can also specify the phase for the calculation.

7. In the **Evaluation Time Step** panel, set how often the solver is to output field solutions. Select **Constant** to output solutions at a preset time interval (e.g., every 0.1 s); or **Number of solve steps** to output solutions each preset number of time steps (e.g., every 3 time steps).
8. Optionally, buttons allow you to **Edit** and **Remove** selected expressions, and to **Remove All** expressions in the table.
9. When finished defining expressions, click **OK** to close the **Solution Setup** dialog box.

Related Topics

Specifying Output Variables

Defining Settings on the Solver Tab for Transient Solutions

To define solver settings on the **Solver** tab of the **Solve Setup** dialog box for transient solutions:

1. Enter a residual value in the **Nonlinear Residual** text box.

   To specify a time-dependent non-linear residual, you can simply type in a function of TIME, such as **sin (TIME)**, or enter an expression that uses a dataset, such as **pwlx(*<name of dataset>*, TIME)**. A time-dependent variable can also be used.

You can save the time-dependent expression as the default for the non-linear residual by clicking the **Save Defaults** button on the **Defaults** tab of the **Solve Setup** dialog box. A new solve setup will then use the user-saved default if the expression can be evaluated. A value of “0” will be used when the expression cannot be evaluated. For example, if the expression depends on a variable that is not yet defined in the design/project.

Changing the definition of the residual will NOT invalidate any existing solution; and the next simulation will continue from the stop time of the last simulation.

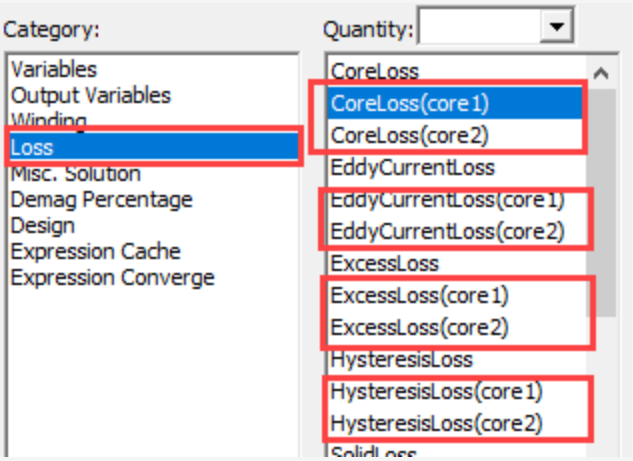
Changing the definition of a *variable* used for the residual value will NOT invalidate any existing solution. The next simulation will be for a different “design variation”.

Changing the definition of a *dataset* used for the residual will NOT invalidate any existing solution. The next simulation will continue from the stop time of the last simulation.

If a variable is used, you can choose to plot a family of curves based on the solved variations of the variable.

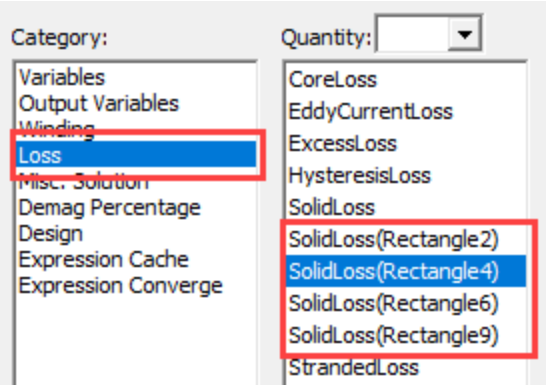
2. For both 2D and 3D Transient designs, you can select the **Smooth BH Curve** check box. This allows the solver to "smoothen" the list of points to make a refined BH curve for better solving. If this option is not selected, the solver just uses the discrete points.
3. Select or clear **Output per object core loss**. When this check box is selected:
   - For steel material, the solver will output core loss for each object, as well as the individual components (hysteresis loss, eddy current loss, and excess loss).
   - For ferrite material, the solver will output only core loss.

Note	To report core loss on objects, the objects must be selected in the **Set Core Loss** dialog box. The loss quantities for the core loss objects will show in the reporter.

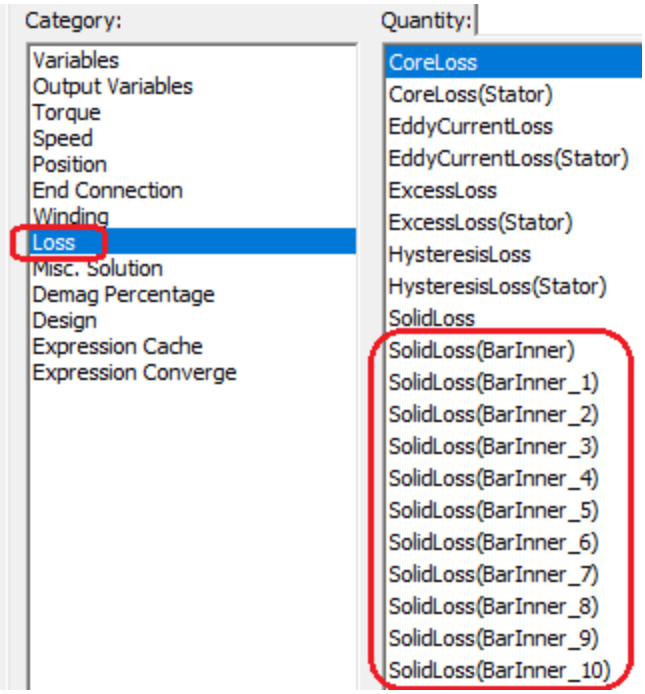


4. Select or clear **Output per object solid loss**. When this check box is selected, solid loss will be output for :
   - The solid loss quantities of the objects selected in the Set Eddy Effects dialog will show in the reporter. (Objects selected should not belong to stranded winding or stranded conductor.)

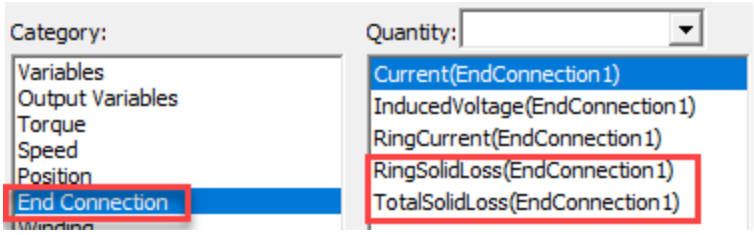
- Coil objects in solid windings.
- For 2D designs:
  - Solid loss quantities of the coil objects assigned with the solid Current Excitation will show in the reporter.



  - Each bar in an end connection will be considered as a separate object for solid loss output.



  - All ring segments in one end-connection will be combined together as one virtual object for solid loss output.



If there are N end-connections (such as double cages), there will be N virtual objects.

5. Select or clear the **Output error** check box. When this check box is selected, the transient solver generates the output error data, which is then sent back to the desktop. The data can be viewed or used in the post processor, just as other transient solution data (such as power loss or winding).

Note	Output error is the percentage error in the total energy of the solution. This is a measure of the accuracy of the finite element approximation and can be used to judge if the mesh is good enough. Output error is calculated by finding the residual energy (the left-over energy calculated from the field after subtracting out the sources) divided by the total energy including the sources. Computing output error uses significant computation time.

6. For 3D Transient analysis only, select the desired **Scalar Potential** shape function: either **Second Order** (default), or **First Order** from the pull-down list. You can also select the desired **Scalar Potential** on the **Solver** tab in the Properties window.

   Use of the First Order scalar shape function, rather than the Second Order shape function, may be desirable for any of the following cases:

   - If computational time is a main concern and solution accuracy is secondary. This might be useful for the initial design stage, before switching to using the more time consuming, and more accurate, second order scalar shape function.
   - If the contour of the geometry is very complicated, it may not be possible to use the higher second order shape function due to the inability of relatively larger element size to properly represent the original geometry shape
   - If the gradient of the magnetic field changes sharply, it may not be possible to use the higher second order shape function with large element size. Using a larger element size may not be able to adequately model the large change of material property with large element size, and nonlinear convergence may be difficult to achieve.

7. For 2D and 3D Transient designs, you can select the **Number of Nonlinear Iterations** check box. This provides an option for you to set both a **Minimum** iteration number to avoid non-converged solution, and a **Maximum** iteration number to avoid taking too much computation time. These values must be integers greater than 0. By default the values for **Minimum** and **Maximum** are 1 and 100, respectively. Maximum number should be greater than the Minimum number.

8. For 2D Transient analysis only, select the algorithm for the **Time Integration Method** using the pull-down list.
   - **Backward Euler** – Default
   - **Runge-Kutta** – third order algorithm for greater accuracy
9. Select or clear the **Steady State** check box.

   For transient simulations whose winding excitations are an AC voltage source, the DC flux linkage component may take a very long time to decay, especially for a device with a large time constant. Selecting **Fast Reach** adds an additional voltage component to the original voltage definition during the first half-cycle to quickly eliminate the DC flux linkage, thus speeding up the solution process. You must also provide a **Frequency of Added Voltage Source** value that is greater than zero. The default value for **Frequency of Added Voltage Source** is 60 Hz. This frequency value is applied to all winding voltage sources.

Note	• This speedup capability is only applicable to voltage sources whose integration of voltage quantity over one period is zero. This means the DC component of all voltage sources is zero. • This feature is intended to be used in conjunction with windings of the type "voltage". If the winding is fed by an external circuit, a similar functionality is available using the alternating flux model defined in the properties of sinusoidal voltage sources. Alternatively, the dedicated **AC_Model** component can be applied if an external circuit is feeding the winding.

10. Select or clear the **Auto Detect** check box.

    This option enables automatic detection of when steady state is reached based on a check of the flux linkage. When this option is selected, you must provide both a **Frequency of Added Voltage Source** value that is greater than zero (default value is 60 Hz), and a **Stop criterion** value that is greater than 0 and less than 1 (default value is 0.005). The smaller the **Stop criterion** value, the longer it will take to auto detect when steady-state is reached. For additional information on auto detect operation, refer to Automatic Detection of Reaching Steady State for Transient Simulations.
11. As needed for your design, select or clear settings in the **Time Decomposition Method Option** panel.

    The Time Decomposition Method (TDM) is an HPC distributed analysis type based on domain decomposition along the time-axis, (rather than the normal geometry division) to parallelize the transient solution. Instead of solving Maxwell transient problems sequentially for each time step, TDM enables time steps to be solved simultaneously in parallel. Thus, TDM can be implemented on distributed memory parallel platforms based on MPI. TDM has very good scalability, resulting in significant speedup for both 2D and 3D transient solutions.

    If you have enabled the **Use Automatic Settings** option in the **Analysis Configuration** dialog box, or if you have enabled the **Transient Solver** distribution type in the **Analysis Configuration** dialog box, you can select:

General Transient	For general transient (Non-Periodic) TDM applications, select **General Transient**. Using this method, TDM solves a collection of

	many time steps simultaneously.
Periodic	TDM method intended for use in steady-state simulation. In such a case, all the setup and solutions at the first time step will be the same as those at the last time step of one period. Setup includes source, boundary, and position in electrical degrees. The solutions include all electrical and magnetic quantities. To use this method, all current in windings and all induced eddy currents in the conducting region should have the same time period. For example, an induction machine cannot be solved using the Periodic method because the time period of the currents in the stator winding is different from the time period of induced eddy current in the rotor bars. Periodical TDM is computationally very efficient if a problem is applicable. But since all time steps over one period have to be solved simultaneously (cannot be divided into subdivision as the case for general TDM transient), the memory requirement may prevent users from using periodical TDM. To this end, the support of half-periodical TDM is able to effectively cut memory usage to roughly half. You can also select a **Half-periodic** option. This TDM option can cut your computer's memory usage when solving Maxwell transient problems. If all physical quantities satisfy anti-periodic condition along Time axis, a problem can be solved using this option so that memory usage can be cut to almost half. Also, since the total time steps to be solved are halved, the total computation time is also cut roughly by half.

Related Topics

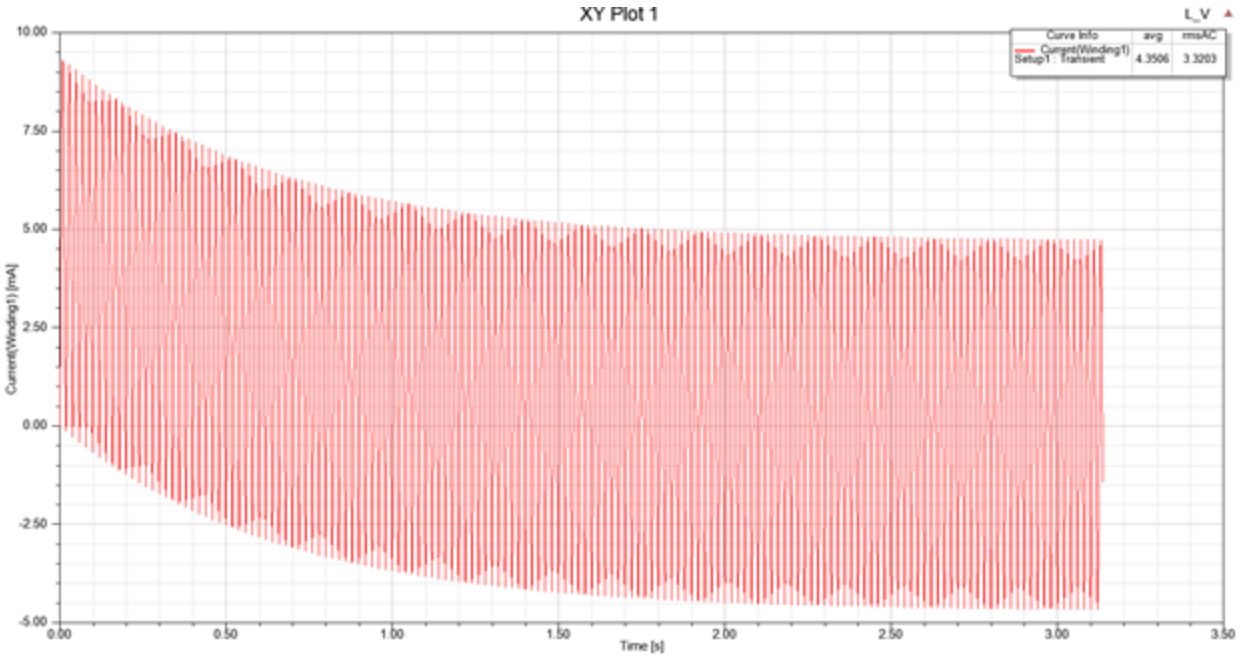
"Time Decomposition Method for Maxwell Transient Designs" on page 16-55

"Automatic Detection of Reaching Steady State for Transient Simulations " below

"Editing Distributed Machine Configurations" on page 21-14

Automatic Detection of Reaching Steady State for Transient Simulations

For transient simulations, when the time constant of the design is large, many cycles may be needed to reach steady state.



Because it is often difficult to predict how many cycles are needed to reach the steady state, the user typically must preset a very long simulation time range (a large stop time) to ensure that steady state can be reached before the end of the simulation. This can result in the waste of significant simulation time for cases in which steady state is reached long before the set stop time.

Enabling the **Auto Detect** setting using a preset **Stop Criterion** causes the solver to check whether steady state has been reached during the simulation, automatically stopping the simulation when steady state is detected. This allows the user to set a very large stop time without concern for wasting needless simulation time, thus improving the efficiency of the simulation without losing the accuracy of the solutions.

Design Considerations

All windings in a project are divided to three categories: AC winding, DC winding, Non-applicable winding.

- For an AC winding, the winding source is periodic with the same frequency as the user-set frequency (**Frequency of Added Voltage Source**) and it has no DC component. A winding which is connected to an external circuit is treated as AC winding.
- For a DC winding the value of the winding source doesn't change throughout the simulation.
- A Non-applicable winding is a winding which cannot be included in either of the first two categories, such as the case when the frequency of the source is different from the frequency set by users.

If a project only includes AC windings and DC windings, automatic detection will be conducted by checking the flux linkages of all AC windings. If a project includes a Non-applicable winding, the automatic detection feature will be disabled.

How Auto Detection Works

Winding information is checked during simulation to determine whether steady state has been reached. If rotational motion is included in the project, the torque is also checked.

Of the three winding parameters that can be used to check whether steady state has been reached during the simulation: *current*, *induced voltage*, and *flux linkage*, **Auto Detect** uses flux linkage for the following reasons:

- Compared with the induced voltage, which is the time derivative of the flux linkage, the flux linkage itself is less sensitive to numerical noise.
- Similarly, current calculations are not as smooth as flux linkage. Also, when the winding excitation is current, the current is not applicable for detecting steady state.

The automatic check for steady state is carried out at the end of each cycle. The cycle time, T is calculated based on the input frequency (**Frequency of Added Voltage Source**) supplied by the user.

$$T=1/f$$

For example, if the check is carried out at time instant t_c, the flux linkage values over the last cycle $[t_c–T, t_c]$ and the second last cycle $[t_c–2T, t_c–T]$ will be used. The first checkpoint, t_c, is at $3T$ since, in general, the first cycle is unsuitable for checking for steady state – especially when **Fast Reach Steady State** is enabled (in which case the first cycle is not even a physical result).

The DC component and the fundamental component of the flux linkage of the two cycles are compared to check for steady state. These components are calculated as follows:

DC component:

$$D = \frac{1}{T}\int_0^T f(t)dt$$

Fundamental component:

$$F = \left|\frac{1}{T}\int_0^T f(t)e^{-2\pi j\frac{t}{T}}dt\right|$$

Assuming the DC component of the flux linkage at the last cycle is D_1, and the fundamental components are denoted by F_1 and F_2 at the last and the second last cycle, the following two criteria are implemented to check for steady state:

$$\left|D_1\right| < r$$

$$\left|F_1 - F_2\right| < r$$

where the reference value r is calculated by:

$$r = |F_1| \times c$$

Coefficient c is the **Stop Criterion** for reaching steady state and is set by the user on the solver tab for transient solutions and must be greater than 0 but less than 1 (default = 0.005). Thus the smaller the coefficient is, the stricter the stop criterion for reaching steady state is, and the longer it will take to auto detect when steady state is reached.

If rotational motion is involved, after the flux linkage has converged, the solver will further check whether the torque has reached steady state. In steady-state, the torque should be constant with small deviation. Such deviation can come from mesh noise, cogging torque, slot effects, etc. This deviation should be small compared to the rms value of the torque. To check whether the torque has reached steady state, the solver compares the difference of the average torques in the last two cycles. The difference should be smaller than the tolerance value. (The tolerance value is normalized by the rms value of the torque in one cycle.)

The average value of the torque and the rms value of the torque in one cycle are calculated as follows:
The average value of the torque:

$$DT = \frac{1}{T}\int_0^T Tor(t)dt$$

The rms value of the torque:

$$RT = \frac{1}{T}\int_0^T Tor(t)^2 dt$$

The following criterion is implemented in the solver to check for steady state for torque:

$$|DT_1 - DT_2| < RT \times 10c$$

As above, *c* is the user-supplied **Stop Criterion.**

Note	Since torque is usually noisier than flux linkage, the tolerance of the torque check is looser than the tolerance of flux linkage.

Related Topics

"Defining Settings on the Solver Tab for Transient Solutions" on page 16-46

Time Decomposition Method for Maxwell Transient Designs

If a Maxwell 2D or 3D transient design problem is too large to solve efficiently on one machine, Maxwell can use the **Time Decomposition Method** (**TDM**). The Time Decomposition Method (TDM) is an HPC distributed analysis type based on domain decomposition along the time-axis (rather than the normal geometry division) to parallelize the transient solution. Instead of solving Maxwell transient problems sequentially for each time step, TDM enables time steps to be solved simultaneously in parallel. Thus, TDM can be implemented on distributed memory parallel platforms based on MPI. TDM has very good scalability, resulting in significant speedup for both 2D and 3D transient solutions.

Maxwell supports two time decomposition methods:

- **General Transient (Non-Periodic)** – This method is intended for general transient applications.

  For this method, TDM solves a collection of many time steps simultaneously – a collection of many time steps being defined as one time sub-division. For example, if the end time of the entire transient simulation is 4 seconds, and if we want to solve all time step solutions together for every 1 second, then this 1 second is defined as one time sub-division, and the total time sub-divisions is 4. For the same problem, we can also define every 2 seconds as one time sub-division. Accordingly, the total time sub-divisions becomes 2.
- **Periodic** – This method is intended for use in steady-state simulation. In such a case, all the setup and solutions at the first time step will be the same as those at the last time step of one period. Setup includes source, boundary, and position in electrical degrees. The solutions include all electrical and magnetic quantities. To use this method, all current in windings and all induced eddy currents in the conducting region should have the same time period. For example, an induction machine cannot be solved using this method because the time period of the currents in the stator winding is different from the time period of induced eddy current in the rotor bars.

  You can also select a **Half-periodic** option if all physical quantities satisfy anti-periodic condition along Time axis. In this case, the memory usage can be cut to almost half.

Periodic TDM has the advantage that the problem just needs to be solved over one period of time, while **General Transient** TDM uses many cycles to reach steady state. But to enforce periodical condition along the Time axis, all time steps over one period of time must be solved together. This may require extensive memory usage.

Using TDM

To use TDM, you must do the following:

- Have and use the HPC license Pool option (Refer to "Setting HPC and Analysis Options for Maxwell and RMxprt Designs" on page 4-54).
- Set the total number of Tasks, N, (at least three Tasks) for the solve pool. The actual number of Tasks for parallel computation is N-1 (dependent tasks), since the first Task is used as the independent task, which is responsible for task assembly, dependent task control, synchronization, etc.

- Enable the **Transient Solver** distribution type in the **Analysis Configuration** dialog box (Refer to "Editing Distributed Machine Configurations" on page 21-14 .)

Note	In Maxwell 2D, for **General Transient** TDM, the expression cache will be updated every 10 subdivisions, that is to say, every (task number - 1)*10 time steps. For **Periodic** TDM, the expression cache is updated after the entire solution is completed.

Limitations to using TDM

- Mechanical transient is not supported. This means all moving positions for every time step have to be known in advance so that FEA will take place at that pre-determined position.
- The solution at a time step cannot depend on the arbitrary history of previous time steps (allowed for previous couple of time step solutions required due to the time integration scheme itself). Traditional core loss computation is supported because core loss computation is done in post-processing.

  Strictly speaking, TDM cannot support hysteresis modeling. But for soft hysteresis materials, especially for lamination core loss computation, the hysteresis loop, or the coercivity Hc, is very small. In such a case, the impact of hysteresis on the nonlinear operating point can be ignored. This means we can decouple the nonlinear iteration process with consideration of hysteresis impact. For the first step of the nonlinear iteration process, distributed parallel TDM can still be applied. After the nonlinear iteration has converged, that is, the nonlinear problem has been linearized for all time steps (or all time steps in the sub-division), the second step is applied to take into account the impact of hysteresis. In such a case, the solution process is sequential – to consider the impact of hysteresis behavior from the previous time-step solution on the solution at current time step. It can be expected that the results with TDM enabled and with TDM disabled might differ slightly due to ignoring the impact of hysteresis on the nonlinear operating point. However, from a practical application point of view, the solutions can be considered as sufficiently accurate –especially for lamination core loss computation.
- The purpose of the demagnetization process in a transient simulation is to find the worst operating point during the entire transient simulation. After a new worst operating point is discovered, a linearized representation of the permanent magnet characteristic will be used to construct a new recoil line, which is applied to the simulation for subsequent time steps. Thus, strictly speaking, TDM cannot support demagnetization modeling because all time steps (or all time steps in a subdivision) are solved simultaneously. However, as a reasonable approximation, we can solve all time steps in one subdivision simultaneously based on distributed parallel TDM – and then search for the worst operating point among the solutions of these just-solved time steps. Thereafter, this new worst operating point is used to construct a new recoil line which is applied to all time steps in the *next* subdivision. This means the impact of previous solutions on subsequent searches for the worst operating point is taken into account subdivision-by-subdivision, rather than time step-by-time step.

  It can be expected that the results of the discovered *final* worst operating point might differ slightly between TDM- enabled solutions and TDM-disabled solutions. However, from a

practical application point of view, the solutions can be considered as sufficiently accurate. This error can be reduced with the use of few time steps in one subdivision.

Related Topics

"Setting HPC and Analysis Options for Maxwell and RMxprt Designs" on page 4-54
"Editing Distributed Machine Configurations" on page 21-14
"Defining Settings on the Solver Tab for Transient Solutions" on page 16-46
Defining Expressions for Transient Solutions

Defining Settings for Output Variables for Transient Solutions

Any output quantity or expression within either the **Transient Report** or **Field Report** type can be defined as an output variable in the **Output Variables** dialog box, accessed via **Results** in the project tree.

To add an output variable to output with time:

- Click **Add** to select the output variables to be added to the list for output with time. The newly defined variable is added to the list.

Note	The **Add** button will not be enabled if there are no variables defined in the project.

To delete an output variable:

- Select the variable from the list, and click **Delete**.

Do one of the following in the **Evaluation Time Step** section on the **Output Variables** tab of the **Solve Setup** dialog box:

- Select **Using Constant**, type a value, and select the units. Select this option if you want to use a constant value for when you calculate output variable values.
- Select **Evaluate every Nth time step**, and type a value. Select this option if you want to evaluate the output variable after a certain number of solution time steps (i.e., instead of calculating the output variable every fixed time interval, you want to calculate it every certain number of time steps). For example, if you want to evaluate the output variable every third time step, type **3** in this text box.

Defining Settings on the Defaults Tab for Transient Solutions

Default analysis parameters are used to initialize the data for new setups.

To set up the default analysis parameters, define the following parameters under the **Defaults** tab of the **Solve Setup** dialog box:

1. Click **Save Defaults** if you want to save the defaults based on the data for the current setup.
2. Click **Revert to Standard Defaults** to clear the existing values.
3. Click **OK**.

Setting Analysis Parameters for Electric Transient Solutions

When you set up an electric transient analysis, define the parameters on the following tabs of the **Solve Setup** dialog box (which appears when you click **Maxwell**>**Analysis Setup>Add Solution Setup**):

- General (Name, Transient Setup)
- Expression Cache
- Solver
- Defaults

Defining Settings on the General Tab for Electric Transient Solutions

The following settings can be defined on the **General** tab of the **Solve Setup** dialog box for transient solutions:

- Setup Name
- Time Steps Setup

Setup Name for Electric Transient Solutions

By default, **Setup1** appears as the name of the first analysis you set up.

To change the name of the setup, type the new name in the **Name** text box on the **General** tab of the **Solve Setup** dialog box.

To disable the setup, uncheck the **Enabled** check box. Refer to *Disabling and Enabling an Analysis Setup* for additional information.

Time Steps Setup for Electric Transient Solutions

To set up a transient analysis, define the following parameters on the **General** tab of the **Solve Setup** dialog box:

1. Enter the **Stop Time**, which is the value at which the solution will stop, and select the units.
2. Enter the **Initial Time Step**, which is the time step used to start the simulation and begin the adaptive process.

   > **Note:** The solver calculates the initial time step based on relaxation time Tau=Eps/Sigma of each material and then compares the results with the initial time step defined by the user. The smaller value is used.

3. Also enter the **Maximum Time Step** to set the allowable range the adaptive process use to alter the time step.
4. Check **Save Fields** to enable saving the fields data.

Defining Expressions for Electric Transient Solutions

You can specify additional evaluation criteria through the use of expressions and output variables.

To set expressions as evaluation criteria:

1. Right-click on the Analysis icon in the Project tree to open the **Solution Setup** dialog box.
2. Click the **Expression Cache** tab and click the **Add...** button to open the **Add to Expression Cache** dialog box.
3. Specify the Context for the expression, including **Report Type**, and **Solution**.
4. Under the **Trace** tab, select from the **Category**, **Quantity**, and **Function** lists to create expressions.

   Selecting a listed category lists the Quantities and Functions available for each category. If you have defined one or more output variables, you can see them by selecting the **Output Variables** category. The **Output Variables** button opens a dialog box that lets you define output variables. See Specifying Output Variables.

   When you have created an expression, it appears in the **Expression** field of the **Trace** tab. If desired, you can use the **Range Function** button to select range functions to apply to the expression.

   Under the **Calculation Range** tab, you can view the values of available sweep variables. Clicking the ellipsis **[...]** button in the **Edit** column opens a list of values from which you can select.
5. When finished defining the expression that you want to add to the cache, click the **Add Calculation** button.

   This adds the selected expression and the associated context to a table in the **Expression Cache** tab. You can define additional expressions with contexts and add them in the same way.
6. When finished adding expressions, click **Done** to close the **Add to Expression Cache** dialog box.

   The **Expression Cache** tab lists the expressions you have added in a table.
   - The **Title** field is editable, by default showing the name as built from the expression (with underscores removed).
   - The **Expression** field shows the full expression. You can resize the **Solution Setup** dialog box, and also each column in the table, to view the contents of the fields.
   - The **Context** column for non-field types lists the parameter type specifying the force, torque or matrix setup. For these types the context also can be given as part of the expression. For example, the expression *Force1.Force_x* with a context of *None* is equivalent to an expression of *Force_x* with a context of *Force1*.

   For field quantities the **Context** can be the geometry on which to evaluate the field quantity - a point or line. If you use a line you need to use some range function to reduce the dimensionality to a single value (or use the Calculation Range tab to specify a single point along the line). Note that, if you create a named expression in the field calculator that already evaluates to a single value, then the **Context** here will likely be *None*.

- The **Intrinsics** column shows a clickable button that opens an **Edit Calculation Range** dialog box. If the **Intrinsics** column button shows **None** you cannot edit the value. If the button shows variables, click the ellipsis **[...]** in the **Edit** column to display a list of the variable values from which you can select. Click **OK** to close the **Edit Calculation Range** dialog box and apply your selections to the **Expression Cache**.

The **Intrinsics** field lists the values you may have set in the **Calculation Range** tab when you added the expression. This is used to set the values of “other” sweeps. Usually there are no other sweeps because the values of the variables are set in the definition of each variation. In some cases you may need to specify the values of “other” sweeps. For field values with a line for the context, you can specify the (normalized) distance along the line For eddy current field quantities you can also specify the phase for the calculation.

7. In the **Evaluation Time Step** panel, set how often the solver is to output field solutions. Select **Constant** to output solutions at a preset time interval (e.g., every 0.1 s); or **Number of solve steps** to output solutions each preset number of time steps (e.g., every 3 time steps).
8. Optionally, buttons allow you to **Edit** and **Remove** selected expressions, and to **Remove All** expressions in the table.
9. When finished defining expressions, click **OK** to close the **Solution Setup** dialog box.

Related Topics

Specifying Output Variables

Defining Settings on the Solver Tab for Electric Transient Solutions

To define solver settings on the **Solver** tab of the **Solve Setup** dialog box for transient solutions:

1. Enter a value in the **Temporal Tolerance**text box.
2. For the Initial Condition, select either a preset **Value** or a **Static Field**.

   If **Static Field** is selected, the **Setup Link** button becomes active, and the **Setup Link** dialog box automatically appears, and the field will be imported from the source design to the target design.
3. The **Import mesh** check box can be selected in the "target" design if the mesh for the same geometry is to be used in the analysis.

   If the **Import mesh** feature is selected, the **Setup Link** button becomes active, the **Setup Link** dialog box automatically appears, and the mesh will be imported from the source design to the target design.

Note	The usual way to create geometrically equivalent designs is by copying all geometry objects from one design and then using **Import From Clipboard** in the other design.

Setup Link for Electric Transient Solutions

If you selected **Import mesh** or **Static Field**, do the following to specify the project, design, and solution containing the mesh information you want to use:

1. Click **Setup Link** in the **Solve Setup** dialog box).

   The **Setup Link** dialog box appears.

2. Click the **General** tab.
3. Select the project from the **Project File** pull-down list.
4. Select the saved design from the **Design** pull-down list.
5. Select the specific solution you want to link to from the **Solution** pull-down list.
6. Click the **Parameters** tab.
7. Optionally, edit the **Value** and **Unit** columns for any of the parameters listed
8. Click **OK** to close the **Setup Link** dialog box and return to the **Solve Setup** dialog box.

Defining Settings on the Defaults Tab for Electric Transient Solutions

Default analysis parameters are used to initialize the data for new setups.

To set up the default analysis parameters, define the following parameters under the **Defaults** tab of the **Solve Setup** dialog box:

1. Click **Save Defaults** if you want to save the defaults based on the data for the current setup.

2. Click **Revert to Standard Defaults** to clear the existing values.
3. Click **OK**.

Resetting Analysis to Time Zero

Resetting the analysis to time zero invalidates the current solution and causes you to start over from the beginning the next time you analyze the simulation.

To reset the analysis, do one of the following:

- Click **Maxwell**>**Analysis Setup>Reset to Time Zero** from the Maxwell menus.
- In the project tree, right-click **Setup** under **Analysis**, and select **Reset to Time Zero** from the shortcut menu.

Changing Memory Settings

The Maxwell solver has to make trade-offs between the solution time and the amount of memory used in the solution process.

The software automatically determines the amount of RAM installed on the computer, assuming that it can use up to 75% of this during the solution process. You can override this choice by changing the following two memory settings:

- Desired RAM Limit
- Maximum RAM Limit

Generally, set the RAM value as high as possible to obtain the fastest solutions.

These settings are not an absolute limit on the amount of RAM to be used. When solving small problems, Maxwell tries to respect this limit by choosing to run more slowly with a smaller amount of RAM. When solving a large problem, Maxwell may be unable to use less RAM than the specified limit. In this case, the amount of memory used exceeds the specified limit but tries to minimize how far it goes beyond.

Specifying the Desired RAM Limit

The **Desired RAM Limit** setting requests a restriction on the amount of physical random access memory (RAM) the MPS solver used in Maxwell may use before it must stop solving *on-core - solving processes entirely in RAM* - and start solving off-core. In *off-core mode*, Maxwell creates temporary solution files to which it spills, or shifts, data from RAM, instead of relying on the operating system to start disk swapping. The location of these temporary files is specified in the General Options (**Tools>Options>General Options**). The MPS solver is finely tuned at handling its own memory, and can optimize loading only those blocks of memory required for its immediate needs.

Using this option may help to keep the entire solver from being swapped out in the normal course of process management on your computer. This kind of control may be especially important when multiple solvers are running on the same machine. Of course, if the total memory requirement of all processes grows large enough, the operating system will be forced into disk swapping.

Note	The Iterative solver doesn't go off-core for efficiency reasons. Because the Desired RAM Limit (the soft memory limit) is used for off-core run, it is not appropriate for the iterative solver. Instead, the iterative solver uses the Maximum RAM Limit (the hard memory limit). Once the limit is exceeded, the iterative solver will issue an "Out of memory" error message.

Regardless of this setting, processes are limited to 4TB of address space on 64 bit operating systems - no matter how much physical memory is installed.

Note	Regardless of the **Desired RAM Limit** setting, if allocation fails, the Maxwell solver automatically switches to off-core mode.

In case you receive an error message regarding insufficient memory on a 64-bit operating system, you may have reached a point where the sum of physical RAM plus available swap space exceeds the minimum amount of RAM needed by the off-core solver. Even for the off-core solver, the RAM usage cannot be made arbitrarily small. In that case you can consider increasing the swap space (the virtual memory) in the settings of your system.

To specify the Desired RAM Limit of the machine on which Maxwell is installed refer to the *HPC and Analysis Options* section.

Specifying the Hard Memory Limit

The **Maximum RAM Limit** setting specifies the absolute limit on the amount of physical memory that the solver can use. If the solver attempts to allocate more memory than this setting, the solution process terminates abnormally, and incorrect error messages may appear.

To specify the **Maximum RAM Limit** for the current user and machine refer to the *HPC and Analysis Options* section.

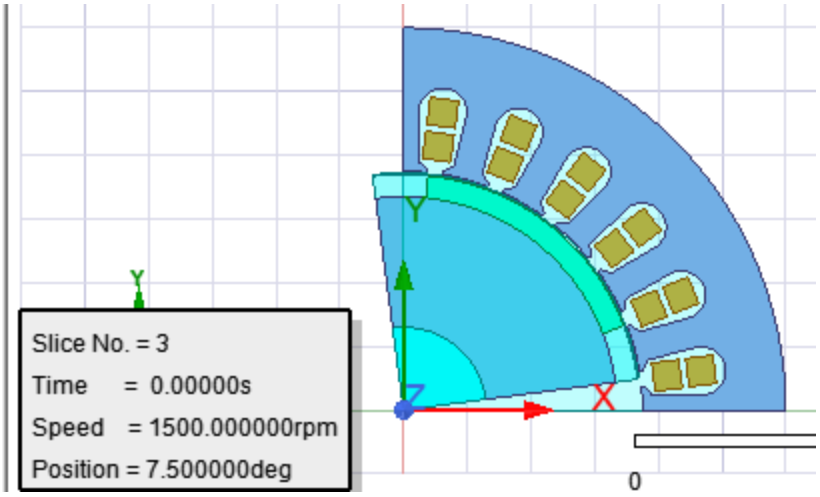
1.

Note	Ansys recommends that you use the **Desired RAM Limit** setting if you want to limit the RAM the solver may allocate.

Setting View Context for Transient Solutions

To set the view context:

1. Click **View>Set Solution Context**.
   The **Set View Context** dialog box appears.
2. Select a solution from the **Solution Name** pull-down list.
3. Select a time instance from the **Time** pull-down list.
4. For Maxwell 2D transient solutions only, in which **Use Skew Model is enabled on the Model Settings tab**, an additional **Slice** field displays, which allows you to select the skew model slice to be displayed in the modeler window.



5. In the **View** area of the dialog, you can choose the location to display the **Solution Context** within the modeler window. **Lower left** was selected for the image above.
6. The position and speed values may be displayed in the **Solution Context** by selecting the **Display Speed and Position values** check box.
7. Click **OK**.

Note	You can also change the view context by double-clicking the solution context text

	label in the bottom left of the model window. **Example of text label showing solution view context:** `Time='0s'`

Speed and Position Display Settings

To set the display setting for speed and position in the view context:

1. Click **View>Set Solution Context**.
   The **Set View Context** dialog box appears.
2. Select the **Motion View Format** tab.
3. In the **Format Speed Unit** section:
   a. Specify the Unit Type from the pull-down list.
   b. Specify the **Field Width** for the display and the number of digits of **Precision** to display.
   c. If you prefer **Scientific Notation**, select the check box.
4. In the **Format Position Unit** section:
   a. Specify the Unit Type from the pull-down list.
   b. Specify the **Field Width** for the display and the number of digits of **Precision** to display.
   c. If you prefer **Scientific Notation**, select the check box.
5. Click **OK**.

Note	You can also change the view context by double-clicking the solution context text label in the bottom left of the model window. **Example of text label showing solution view context:** `Time='0s'`

Renaming a Solution Setup

Do the following to rename a solution setup:

1. In the project tree, under **Analysis**, right-click the setup you want to rename.
   A shortcut menu appears.
2. Select **Rename** from the shortcut menu.
   The setup name text is highlighted in the project tree.
3. Type the new name for the setup, and press **Enter**.

You can also rename the solution setup by changing the text in the **Name** text box of the **Solve Setup** dialog box.

Related Topics

Copying a Solution Setup

Copying a Solution Setup

Solution setups may be copied and pasted within a design or across designs of the same type. This is beneficial for setups having a large number of parameters to specify such as an RMxprt design, or where minor changes to a setup are being evaluated.

Note	Setups cannot be pasted across solution types, across geometry modes such as from 3D to 2D, or across machine types in RMxprt.

Do the following to copy a solution setup:

1. In the project tree, under **Analysis**, right-click the setup you want to copy.

   A shortcut menu appears.
2. Select **Copy** from the shortcut menu.

   The setup parameters are copied to the clipboard.
3. In the project tree, right-click on the **Analysis folder to receive the copied setup**.

   A shortcut menu appears.
4. Select **Paste** from the shortcut menu.

   The setup parameters are copied to the **Analysis** folder as a new setup.

Related Topics

Renaming a Solution Setup

Disabling and Enabling an Analysis Setup

To disable an **Analysis** setup definition without deleting it:

1. Expand the tree hierarchy under the Analysis icon in the project tree.
2. Right-click on the icon for the setup definition. In the shortcut menu that appears, click Disable Setup.

   You can also disable an analysis setup by selecting **Properties** from the shortcut menu to open the **Edit Properties** window. In the upper right corner of the Edit Properties window, uncheck the **Enabled** box. Then click **OK**.

To reactivate an **Analysis** setup:

1. Expand the tree hierarchy under the **Analysis** icon in the project tree.
2. Right-click on the icon for the setup definition. In the shortcut menu that appears, click **Enable Setup**.

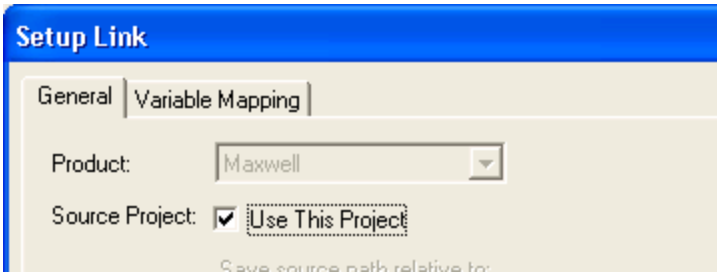
   You can also reactivate the analysis setup by selecting **Properties** from the shortcut menu for the sweep to open the **Edit Properties** window. Check the **Enabled** box in the upper right corner, and click **OK** to apply the change and close the window.

Setup Link Dialog

The **Setup Link** dialog permits you to link the current project to another as described in the following topics:

- Setup Link for Non-Transient Solutions
- Import Mesh for Non-Transient Solutions
- Setup Link for Transient Solutions
- Import Mesh for Transient Solutions
- Setup Link for Electric Transient Solutions

The **General** and **Variable Mapping** tabs are common to all Setup Link dialog boxes.



Depending on the type of link being set up, other tabs such as **Additional Mesh Refinements**, and Advance may also appear. Details of the use of the various tab settings may be found in the topics linked above.

Clearing Linked Data

If you have previously setup links, the **Maxwell>Analysis Setup** menu contains an option to **Clear Linked Data**. This removes the linked data for all links in a target design, therefore invalidating the solutions. Clearing linked data for some link types requires Maxwell to revert to the initial mesh. Thus, in some cases, this command removes the current mesh of the target design.

Setting the Report Updating During Analysis Options

Updating numerous reports may take a significant amount of time. Updating reports during the analysis process can impact the overall time to solution. You may want to vary the times when your reports get updated relative to the impact on overall solve time.

Four options exist for updating reports during solutions:

- After each variation - when performing an Optimetric or parametric analysis, all reports are updated after analysis of each variation has been completed.
- Automatically - the default. It means update most things immediately.
  For "AdaptivePass" plot context, plots are updated at the end of each solution pass. For "LastAdaptive" or "Transient" the plot is updated at the end of the transient or adaptive solution.
  This option balances report and field plot updating with solution time. For example, reports

may be updated after each adaptive pass but field plots will not be updated until the solution is complete.

- Immediately - update reports and plots as soon as data comes from the solver.
  This option will have the greatest impact on the overall solution time but will have the most rapid updating of reports and field plots. Caution should be used in selecting this option. Some types of reports and field plots may take a long time to update, especially as the mesh size increases.
- Never - only manual intervention updates reports.
  This option will prevent updates from impacting the solution time.
- On Completion - as with Never, but a single update is done when the solve completes. This option should be considered when using dynamic links to other products to provide best solution time with the final data available for the link.

To change the setting for the current project, see Desktop Performance options.

17 - Maxwell Coupling

Coupling is provided in specific contexts between Maxwell and the following products:

- **Ansys Workbench**

  Coupling of Maxwell2D/3D with Ansys Workbench is supported via the Workbench schematic. The following types of coupling with Workbench are supported:

  - Thermal feedback is supported for Maxwell magnetostatic, eddy current, and transient solution types.
  - Stress feedback is supported for Maxwell magnetostatic, eddy current, and electrostatic solution types.
  - Harmonic force is supported for Maxwell transient and Eddy current solutions.
  - Partial Model Coupling

  For more information, see “Maxwell to Ansys Thermal Coupling” and “Maxwell to Ansys Stress Feedback Coupling,” and "Coupling Maxwell with Both Ansys Thermal and Structural via Workbench."

- **HFSS**

  There are two possible coupling mechanisms between HFSS and Maxwell:

  - Near field coupling.
  - Magnetic bias source coupling.

  While they serve different application objectives, they are similar in neither requires the geometry in the coupled designs to be identical.

  For more information, see “Near Field Coupling between Maxwell and HFSS” and "Magnetic Bias Source Coupling between Maxwell and HFSS."

- **Twin Builder**

  Coupling between Maxwell and Twin Builder can take the following forms (initiated through the Twin Builder user interface):

  - Export a parametric matrix from Maxwell to an ECE model file for placement in Twin Builder.
  - Dynamically link a Maxwell capacitance or inductance project to a component in Twin Builder.
  - Dynamically link a Maxwell Transient design to Twin Builder to perform Transient-Transient cosimulation.
  - Dynamically link a Maxwell design to Twin Builder to perform State Space cosimulation.
  - Dynamically link to RMxprt through Dynamic SML.

  Because the thermal and/or stress solutions are derived from the distributed power loss and force density, they are likely to be more accurate than analyses based on average losses/force.

  For more information, see "Coupling Maxwell to a Twin Builder Component."

- **Ansys Fluent**

Coupling without using Ansys Workbench is supported between Ansys Fluent and Maxwell 3D magnetostatic and Eddy current designs in Maxwell. A transient Ansys Fluent simulation can use a Maxwell solution to simulate the magnetic effect in the electric arc of contactors.

For more information, see "Coupling Maxwell Magnetostatic and Eddy Current Designs with Ansys Fluent."

- **LMS Virtual.Lab™**

  Maxwell supports one-way coupling with Siemens® LMS Virtual.Lab™.

Related Topics

Maxwell to Ansys Thermal Coupling
Setting the Temperature of Objects
Maxwell to Ansys Stress Feedback Coupling
Maxwell to Ansys Harmonic Force Coupling
Maxwell Multiple RPM Harmonic Force Coupling
Near Field Coupling between Maxwell and HFSS
Magnetic bias source coupling between Maxwell and HFSS
Coupling Maxwell to a Twin Builder Component
Coupling a Maxwell Magnetostatic or Eddy Current Design with Ansys Fluent

Maxwell to Ansys Thermal Coupling

Coupling Maxwell2D/3D with Ansys applications is supported via the Workbench schematic. Thermal feedback is supported for Maxwell magnetostatic, eddy current, ac and dc conduction, and transient types. Users also need to set up the design and geometry appropriately. An appropriate design should be temperature-dependent, and have one or more solve setups that are enabled for thermal feedback.

Note	A Maxwell system can also be coupled with both thermal and stress systems via Workbench.

Maxwell-Specific Considerations

- If there is thermal feedback, each frequency point in an eddy current sweep will be solved with the same feed-back data.
- Temperature distribution data that has already been exported from Ansys Thermal will be invalidated when users continue the Maxwell solution in a way such that the Maxwell mesh is being regenerated.
- Once the Maxwell solution has been integrated with a thermal effect, the adaptive process of a static simulation will be disabled. Users must then manually invalidate the temperature data in the existing solution before the adaptive process will be re-enabled.
- Maxwell solutions can incorporate feedback from both thermal and stress.

Maxwell 2D Coupling Considerations

For Maxwell2D in RZ mode, geometry either must be exported into the XY plane, or the geometry must be rotated into the XY plane after it is imported into **Ansys Mechanical**. This is because Mechanical 2D requires geometry to be in XY plane.

There are two modes in Ansys Thermal 2D. The default mode is called **Planed Stress** where zero stress and non-zero strain are assumed in the z-direction. This is similar to the XY mode in Maxwell 2D. The other mode is called **Axisymmetric**. This mode will be used when a 3D model can be generated by revolving a 2D section 360 degree. This is similar to the RZ mode in Maxwell 2D. Users must ensure that they use the correct geometry mode in Thermal 2D. Invalid geometry mode selection is prohibited by the Thermal2D interface.

There are two types of **Import Load** in Mechanical: **Heat Generation** and **Heat Flux**. **Heat Generation** is used when mapping losses from *objects* in Maxwell2D/3D and **Heat Flux** should be used to map the loss from the *edges* of objects in Maxwell 2D/3D. **Heat Flux** is supported only for eddy current designs with impedance boundary. Users can insert multiple **Heat Generation** or **Heat Flux** loads via the **Imported Load (Maxwell2D/3DSolution)** objects.

Related Topics

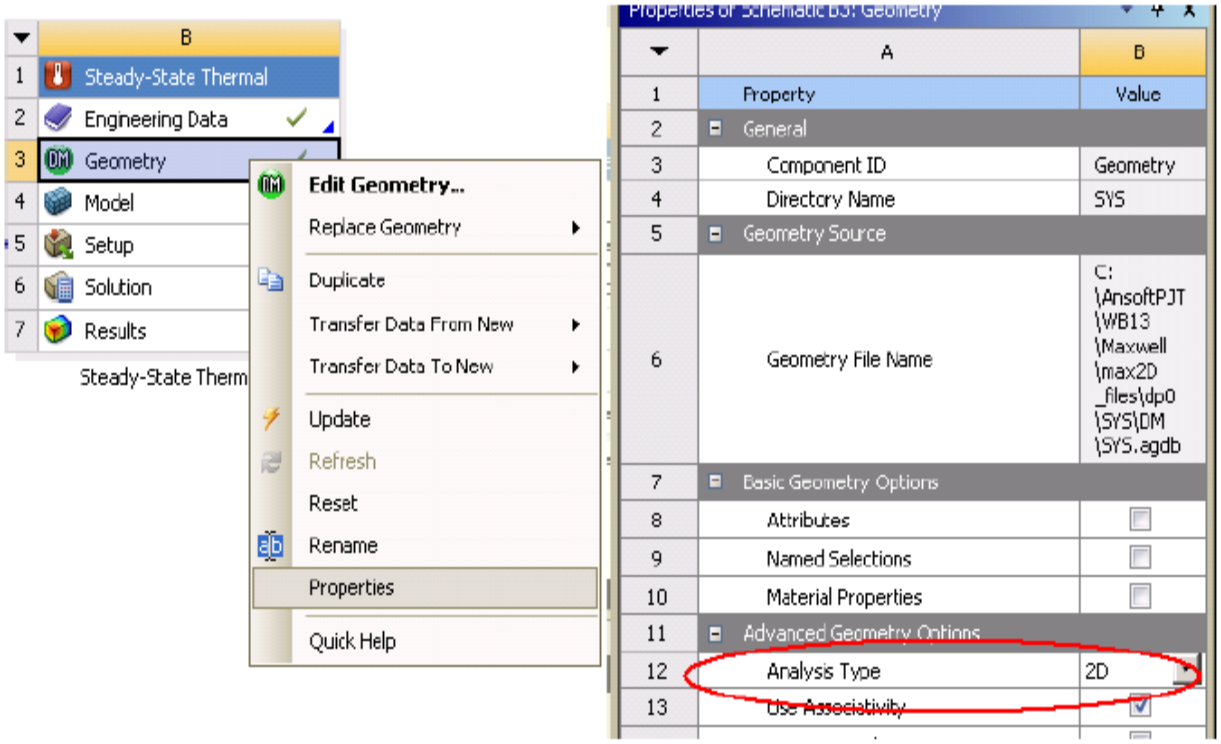
Coupling Maxwell Designs with Ansys Thermal via Workbench

Setting The Temperature of Objects

Coupling Maxwell Designs with Ansys Thermal via Workbench

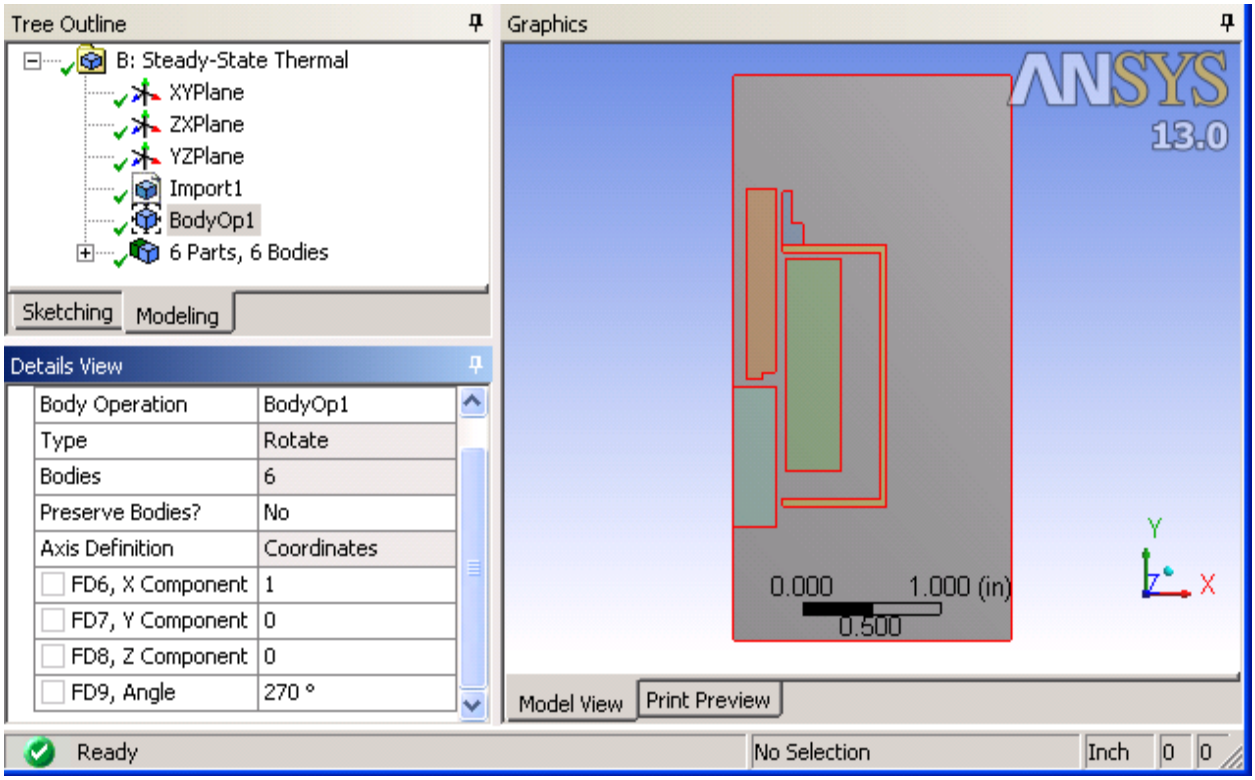
Coupling Maxwell2D/3D with Ansys applications is supported via the Workbench schematic. Thermal feedback is supported for Maxwell magnetostatic, eddy current, ac and dc conduction, and transient types. Users also need to set up the design and geometry appropriately. An appropriate design should be temperature-dependent, and have one or more solve setups that are enabled for thermal feedback.

1. The easiest way to add a Maxwell 2D or 3D design to a Workbench schematic is to import a working design via **WorkbenchFile>Import**. The imported design is placed in the **Workbench** schematic after it is successfully imported.
2. Next, insert a **Steady-State Thermal** system and change its **Analysis Type** to **2D** or **3D**, (depending on the Maxwell design type) by right clicking on the **Geometry** cell and selecting **Properties**. It is important to change the **Steady-State Thermal** system's analysis type *before* setting up its geometry.

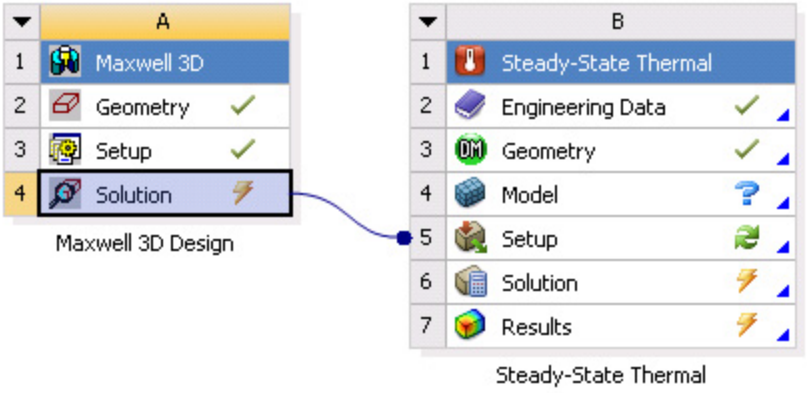


3. to set up the **Steady-State Thermal** system's geometry, you must first export the Maxwell geometry using **sat** or **step** format as follows:
   a. Select the **Modeler>Export** menu item.
   b. Select the desired model geometry format (**sat** or **step**), and the save location in the dialog box and save the file for use by Ansys Workbench.
4. Import the file via the **Geometry** module of the **Steady-State Thermal** system.
   a. To access the **Geometry** module, double click on the **Geometry** cell in the **Steady-State Thermal** system to launch **DesignModeler**.
   b. Select **File>Import External Geometry File** and browse for the geometry file exported from Maxwell.
   c. After the geometry file is imported, right-click on the root folder of the modeler project tree and select **Generate**. (When the geometry file is of a Maxwell 2D RZ design, users

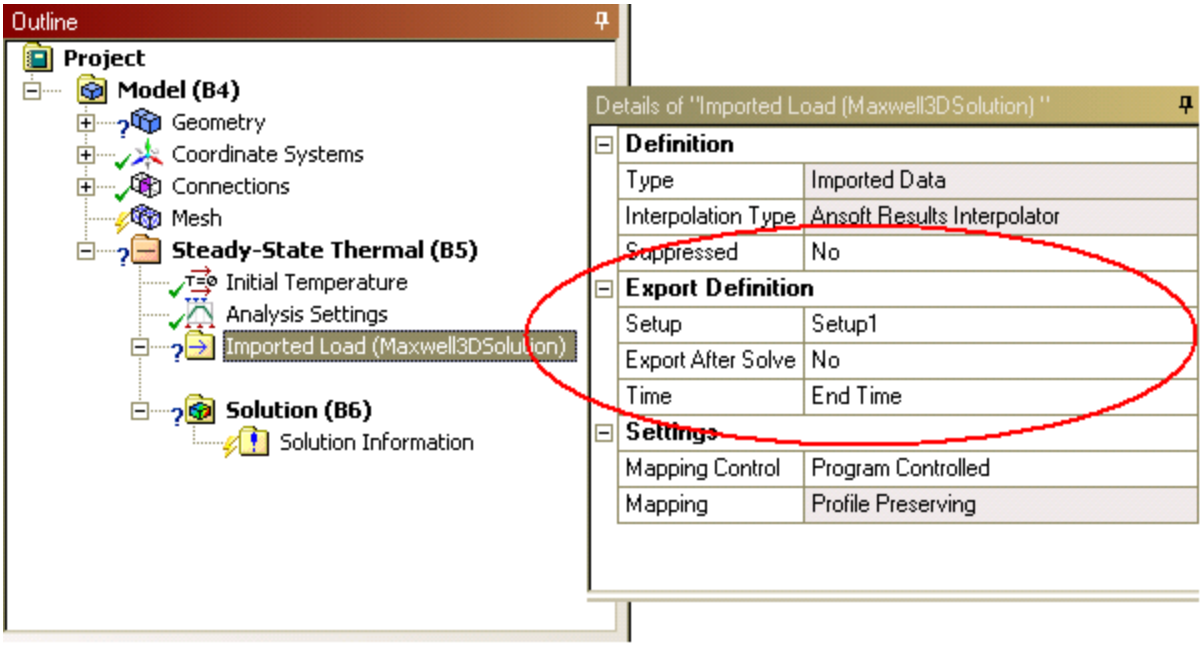
can rotate the geometry in **DesignModeler** by creating a body operation.)



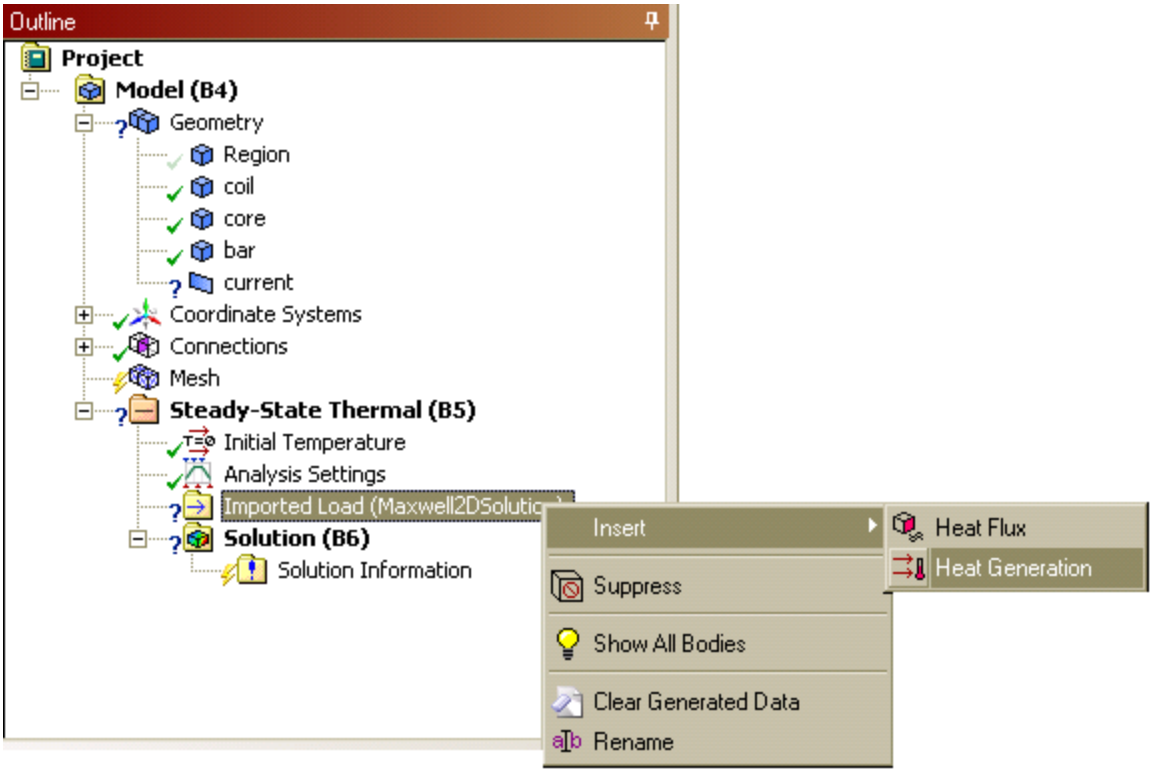
5. Close **DesignModeler** and refresh the **Model** cell of the **Steady-State Thermal** system by right-clicking the **Model** cell and selecting **Refresh**.
6. The geometry mode of the **Steady-State Thermal** system can be changed via the Ansys **Mechanical** user interface.
   a. Launch **Mechanical** by double clicking the **Setup** cell of the **Steady-State Thermal** system.
   b. Select **Geometry** in the project tree and the **Definition** of **Geometry** will be shown in the **Detail** window.
   c. Select either **Plane Stress** or **Axisymmetric** as the value for the property 2D (or 3D) **Behavior**.
7. To set up the coupling, drag the **Solution** cell of the Maxwell system and drop it on the **Setup** cell of the **Steady-State Thermal** system.



8. Note that the Maxwell **Solution** cell is tagged with a “Lighting Bolt” symbol. Right-click on the Maxwell **Solution** cell and select **Update**. This will initiate a Maxwell simulation if it is not already solved. Once Maxwell's solution is available, the “Lighting Bolt” changes to a “Green Check”symbol.
9. To “push” the coupling into **Steady-State Thermal**, right-click on the **Steady-State Thermal Setup** cell and select **Refresh**.
10. After refresh is finished, you can launch Ansys **Mechanical** by double-clicking the **Setup** cell to finish the coupling setup.
11. In the Ansys **Mechanical** application project tree, an **Imported Load (Maxwell2DSolution)**, or **(Maxwell3DSolution)**, item should already be inserted. Select the **Imported Load** folder to view its details. Because the inserted Maxwell2D (or 3D) system supports thermal feedback, the **Details** window shows information regarding how the temperature result should be exported, and what type of mesh mapping should be used.

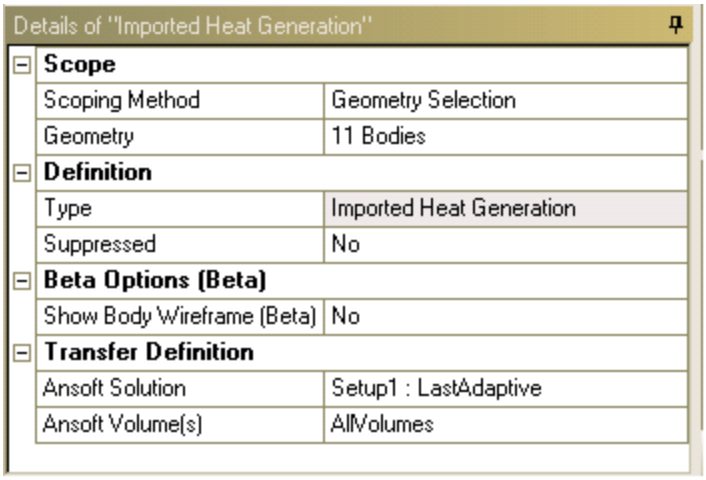


12. To finish the coupling setup, you must insert either an **imported Heat Generation** or an **imported Heat Flux** boundary condition. **Heat Generation** is used when mapping loss from objects in Maxwell; and **Heat Flux** should be used to map the loss from the edges of objects in Maxwell. Users can insert multiple **Heat Generation** or **Heat Flux** loads via the **Imported Load (Maxwell2DSolution)**, or **Imported Load(Maxwell3DSolution)** objects.
13. To insert a **Heat Generation**, use **Body** select by clicking the  icon in the **Mechanical Toolbar**. Then click on the objects where the EM loss should be imported. After all the desired objects are selected, right-click on **Imported Load (Maxwell2DSolution)> Insert > Heat Generation**, or **Imported Load(Maxwell3DSolution) > Insert > Heat Generation**.



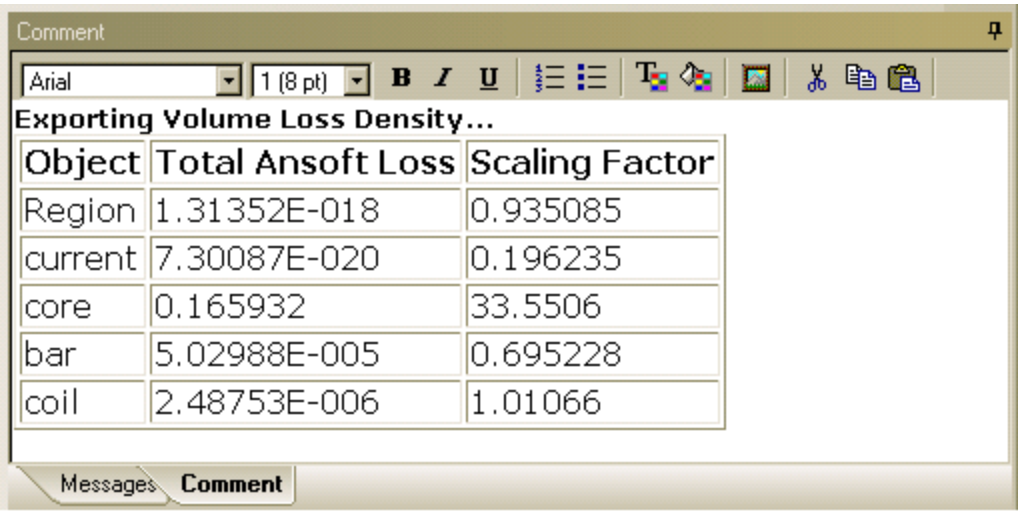
A sub-item named **Imported Heat Generation** will appear in the project tree.

14. Click on the **Imported Heat Generation** tree item to view its details.

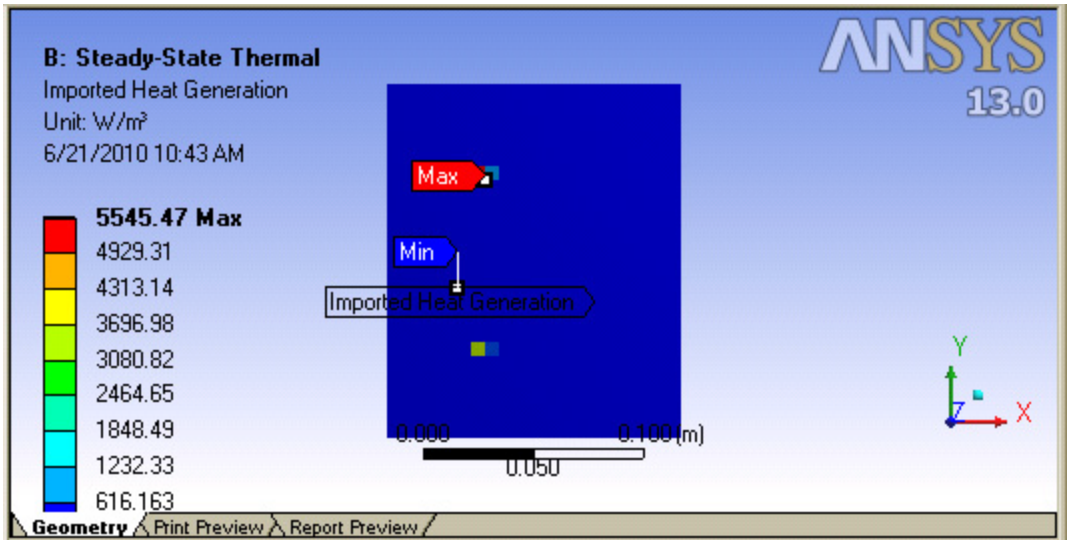


15. In the **Transfer Definition** section, users can set up the source Maxwell solution by pulling down the **Ansoft Solution** combo box and select one of the listed Maxwell solutions.
16. Right-clicking on **Imported Heat Generation > Import Load** will import loss from the **Ansoft Solution** selected for this load. After import has completed, the **Import Heat Generation** item becomes a folder, and an entry called **Imported Load Transfer**

Summary is listed. Select the **Imported Load Transfer Summary** entry and the scaling factors used to export the load from Maxwell will be displayed in the **Comment** window.



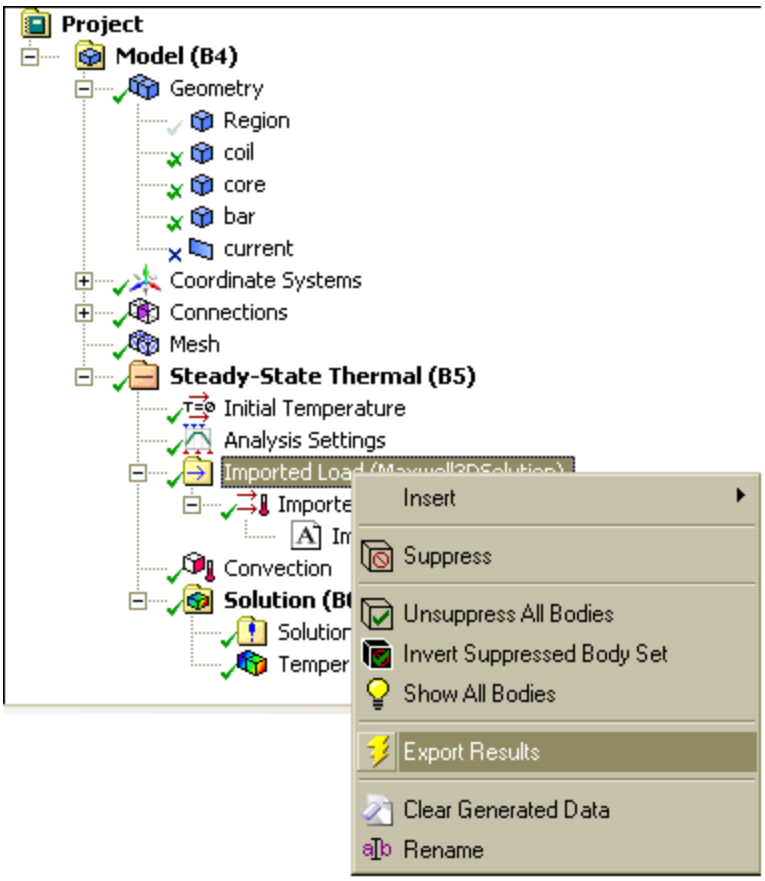
17. Selecting **Import Heat Generation** should show an overlay-plot of the imported load. The loss mapping from Maxwell can be verified by comparing this overlay-plot with an Ohmic-Loss field overlay plot in Maxwell.



18. Create a **Convection** boundary to complete the thermal setup. Use **Edge** select by clicking the  icon at the **Mechanical Toolbar** and then **Edit > Select All**. With the edges selected, right-click on the **Steady-State Thermal** project tree item and insert a **Convection**. With the **Convection** item selected, change its **Film Coefficient** to **5 W/m2** via the **Detail** window. Right-click on the **Solution** tree item and select **Solve**. After the thermal solution is finished, users can insert a **Temperature** plot by right-clicking on

Solution and selecting **Insert > Thermal > Temperature**. Right-click on the newly inserted **Temperature** item and select **Evaluate All Results**.

19. To export the thermal result to Maxwell, right-click on the **Imported Load (Maxwell2DSolution)**, or **Imported Load(Maxwell3DSolution)** and select **Export Results**.



20. To fully utilize the automation capabilities provided in **Ansys Workbench**, select **Imported Load (Maxwell2DSolution)**, or **Imported Load(Maxwell3DSolution)**; and in its **Detail** window, select **Yes** for **Export after Solve**. With this option selected, users can continue the iteration between Maxwell/Thermal simulations from the Workbench schematic.

    To “push” the exported thermal results back to Maxwell, right-click on Maxwell's **Solution** cell on the Workbench schematic and select **Enable Update**. Then, right-click again on Maxwell's **Solution** cell and select **Update**. This will trigger Maxwell to re-simulate its solution with thermal results.

    To continue the solve iterations, repeat the following steps as needed:

    a. Right-click on Thermal's **Setup** cell and select **Refresh**.
    b. Right-click on Thermal's **Setup** cell and select **Update**.
    c. Right-click on Maxwell's **Solution** cell and select **Enable Update**.
    d. Right-click on Maxwell's **Solution** cell and select **Update**.

Maxwell to Ansys Stress Feedback Coupling

Maxwell 2D/3D supports stress feedback coupling for non-transient designs via the Workbench schematic. Any static design type that can support one-way stress coupling can support stress feedback. For both 2D and 3D, the supported design types are: magnetostatic, eddy current, and electrostatic.

Note	A Maxwell system can also be coupled with **both** thermal and stress systems via Workbench.

Maxwell-Specific Considerations

- Users must select object/s (i.e., geometry) in Maxwell upon which displacement data from Ansys Mechanical should be considered. This is done to avoid applying displacement to the whole model while only certain part(s) are sensitive to stress. This is a per-design setting similar to the Set Object Temperature setting for thermal feedback.
- Each frequency point in an eddy current sweep will be solved with the same feedback data.
- Displacement data that has already been exported from Ansys Mechanical Stress is invalidated when the Maxwell design is edited such that the previously simulated solution is considered as invalid or non-current. These edits include:
  a. Modifying the geometries or selecting “Revert To Initial Mesh.”
  b. Modifying the boundary/excitation settings, such as changing the value of a “Current excitation” in a magnetostatic design.
- Maxwell solutions can incorporate feedback from both thermal and stress.
- Once the Maxwell solution has been integrated with displacement/temperature, the adaptive simulation process of a static simulation is disabled. Users must then manually invalidate the displacement/temperature data in the existing solution before the adaptive simulation process is re-enabled.
- Simulation iteration is controlled by user.

Related Topics

Coupling Maxwell Designs with Ansys Structural via Workbench

Setting the Deformation of Objects

Maxwell to Ansys Thermal Coupling

Force Densities in Maxwell

Force Density Calculation in Maxwell

Maximum force density (AC) components calculation

In Maxwell, there are two options to couple the eddy solver with the Mechanical structure static solver. You can choose either DC force density or Maximum force density. The maximum force density includes the contribution from AC components.

In Maxwell stress tensor,

$$\sigma_{ij} = B_i B_j - \frac{1}{2\mu_0} B^2 \delta_{ij}$$

The DC(averaged) components of B_iB_j is calculated by

$$B_i B_j\big|_{DC} = \frac{1}{2\mu_0} Re\left[\dot{B}_i \dot{B}_j\right]$$

and the AC (fluctuation) components

$$B_i B_j\big|_{AC} = \frac{1}{2\mu_0} Re\left[\dot{B}_i \dot{B}_j\right]$$

where

$$\dot{B} = B_{Re} + jB_{im}$$

And $\dot{B}$ is the complex conjugate of $\dot{B}$. The maximum force is calculated as:

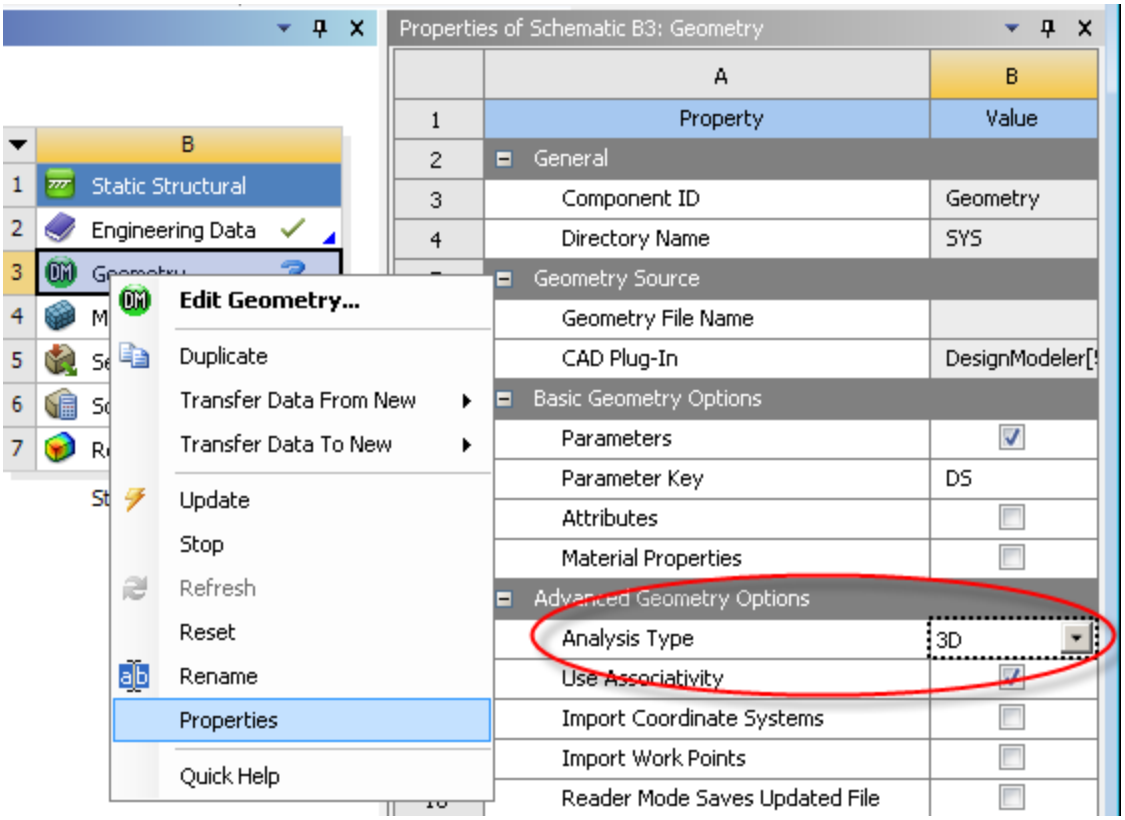
$$\vec{F} = \frac{\vec{F}_{DC}}{|\vec{F}_{DC}|}\left(|\vec{F}_{DC}| + |\vec{F}_{AC}|\right)$$

Coupling Maxwell Designs with Ansys Structural via Workbench

Stress feedback coupling between Maxwell2D/3D and ANSYS Structural 16 is supported via the Workbench schematic. Stress feedback is supported for Maxwell magnetostatic, eddy current, and transient types. Users also need to set up the design and geometry appropriately. An appropriate design has one or more solve setups that are enabled for stress feedback. The process for stress feedback coupling between Maxwell and Ansys Structural is similar to that described in "Coupling Maxwell Designs with Ansys Thermal via Workbench".

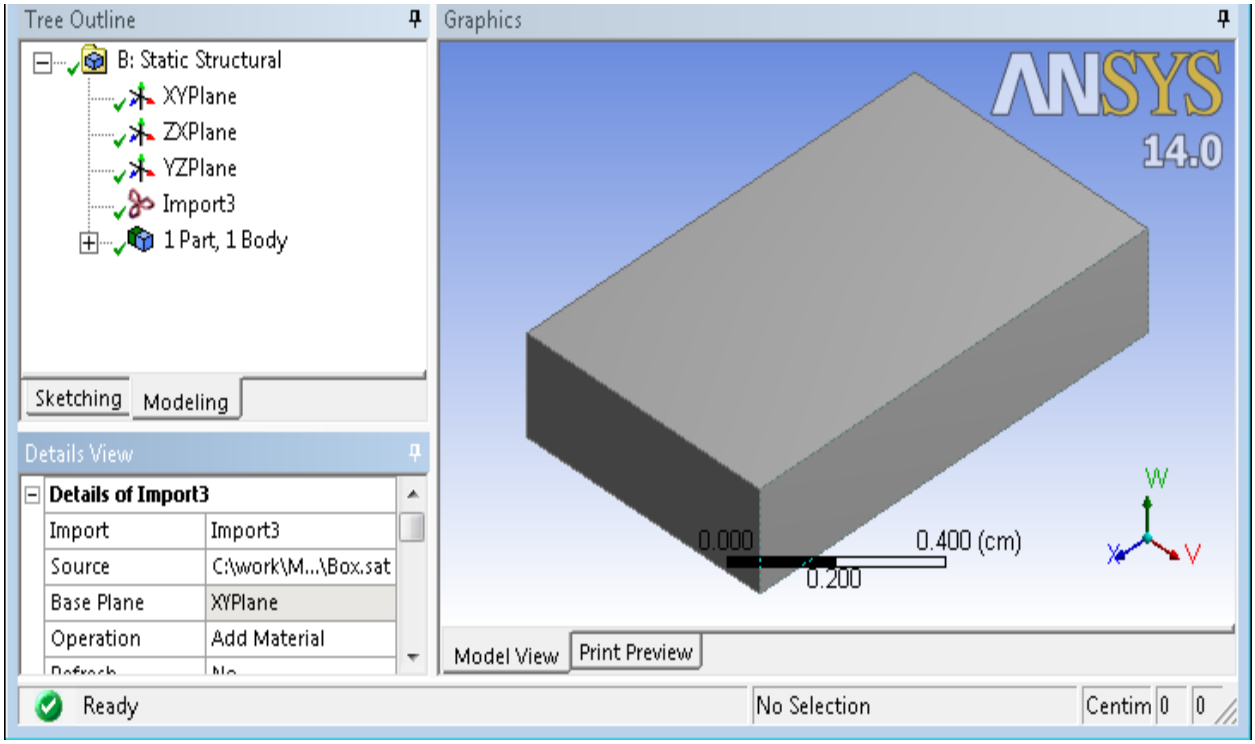
1. The easiest way to add a Maxwell 2D or 3D design to a Workbench schematic is to import a working design via **WorkbenchFile>Import**. The imported design is placed in the **Workbench** schematic after it is successfully imported.
2. Next, insert a **Static Structural** system and change its **Analysis Type** to **2D** or **3D**, (depending on the Maxwell design type) by right clicking on the **Geometry** cell and

selecting **Properties**. It is important to change the **Steady-State Thermal** system's analysis type *before* setting up its geometry.

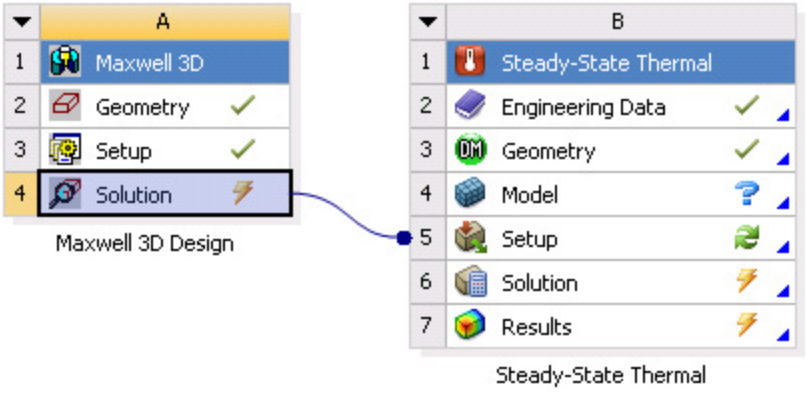


3. to set up the **Static Structural** system's geometry, you must first export the Maxwell geometry using **sat** or **step** format as follows:
   a. Select the **Modeler>Export** menu item.
   b. Select the desired model geometry format (**sat** or **step**), and the save location in the dialog box and save the file for use by Ansys Workbench.
4. Import the file via the **Geometry** module of the **Static Structural** system.
   a. To access the **Geometry** module, double click on the **Geometry** cell in the **Static Structural** system to launch **DesignModeler**.
   b. Select **File>Import External Geometry File** and browse for the geometry file exported from Maxwell.
   c. After the geometry file is imported, right-click on the root folder of the modeler project tree and select **Generate**. (When the geometry file is of a Maxwell 2D RZ design, users

can rotate the geometry in **DesignModeler** by creating a body operation.)



5. Close **DesignModeler** and refresh the **Model** cell of the **Static Structural** system by right-clicking the **Model** cell and selecting **Refresh**.
6. The geometry mode of the **Static Structural** system can be changed via the Ansys **Mechanical** user interface.
   a. Launch **Mechanical** by double clicking the **Setup** cell of the **Static Structural** system.
   b. Select **Geometry** in the project tree and the **Definition** of **Geometry** will be shown in the **Detail** window.
   c. Select either **Flexible** or **Rigid** as the value for the property 2D (or 3D) **Stiffness Behavior**.
7. to set up the coupling, drag the **Solution** cell of the Maxwell system and drop it on the **Setup** cell of the **Static Structural** system.



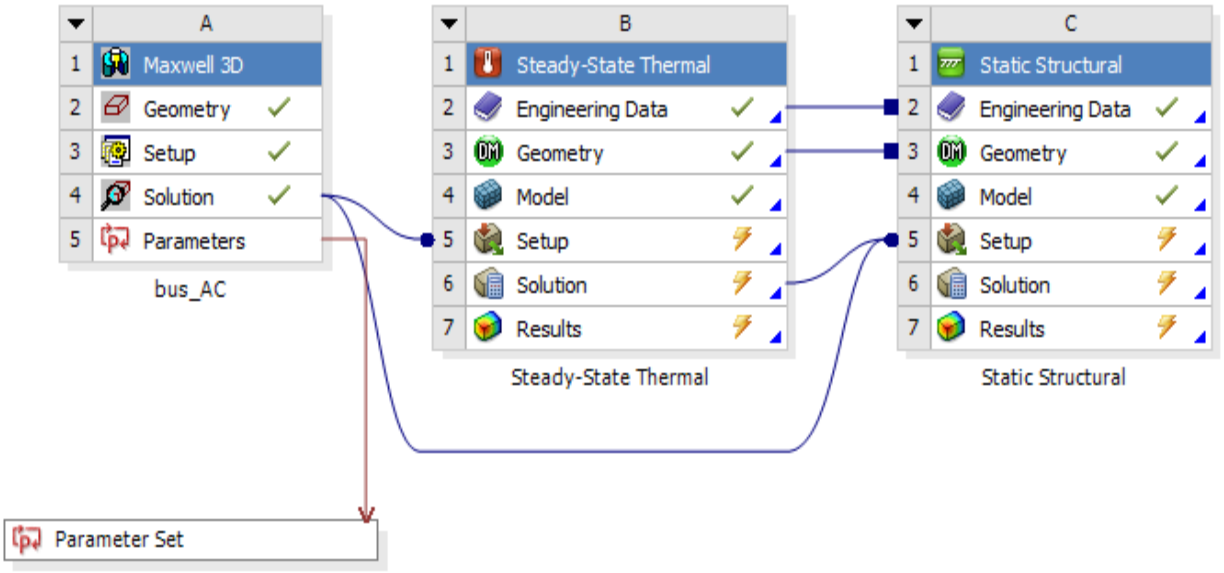
8. Note that the Maxwell **Solution** cell is tagged with a “Lighting Bolt” symbol. Right-click on the Maxwell **Solution** cell and select **Update**. This will initiate a Maxwell simulation if it is not already solved. Once Maxwell's solution is available, the “Lighting Bolt” changes to a “Green Check”symbol.
9. To “push” the coupling into **Static Structural**, right-click on the **Static Structural Setup** cell and select **Refresh**.
10. After refresh is finished, you can launch Ansys **Mechanical** by double-clicking the **Setup** cell to finish the coupling setup, which is similar to that described in “Coupling Maxwell Designs with Ansys Thermal via Workbench”.

stress Stress

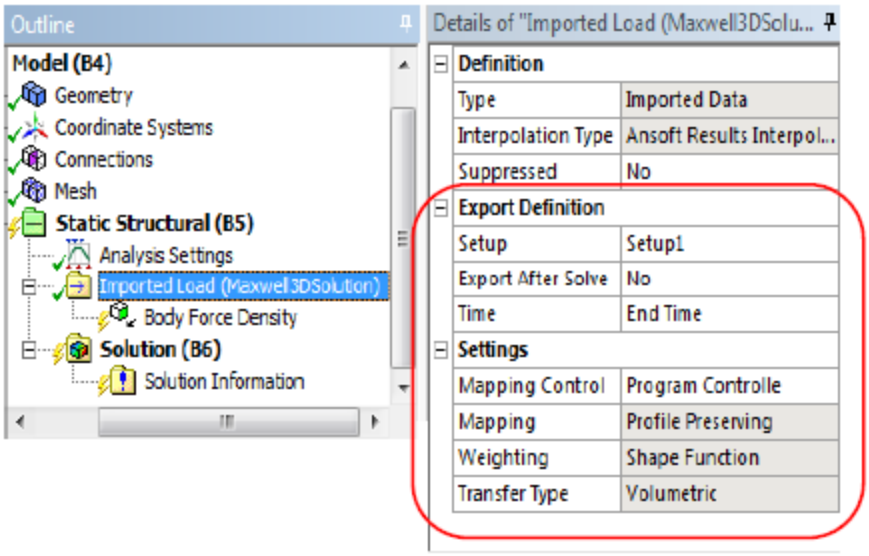
Coupling Maxwell with Both Ansys Thermal and Structural via Workbench

A Maxwell system can also be coupled with *both* thermal and stress systems via Workbench. The thermal system then serves as an “up-stream” system for the stress system. This means that the stress analysis takes both electromagnetic and thermal forces into consideration. In this scenario the user must select the same object(s) in the thermal and stress systems to apply the respective imported loads.

to set up the coupling, drag the “Solution” cell of the Maxwell system and drop it at the “Setup” cell of a Thermal and Static Structural (Stress) system. The user also needs to couple the Thermal and Static Structural systems to capture the effect of thermal force.

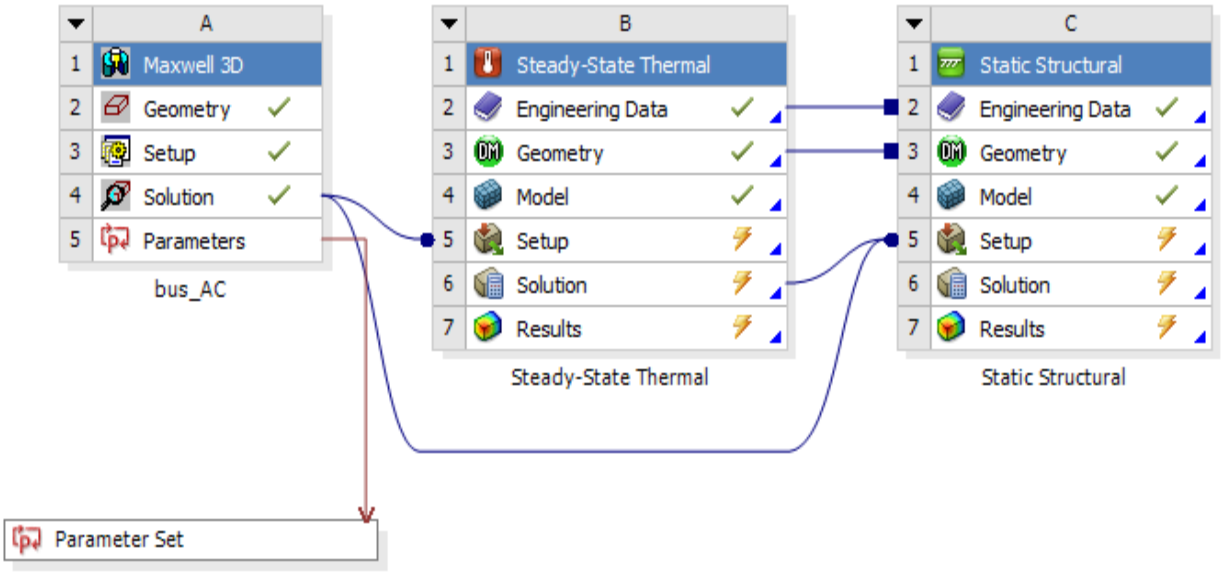


The coupling framework is built on top of the existing Thermal Feedback. The Stress system will have additional Export Definition and Mapping Settings like Thermal.



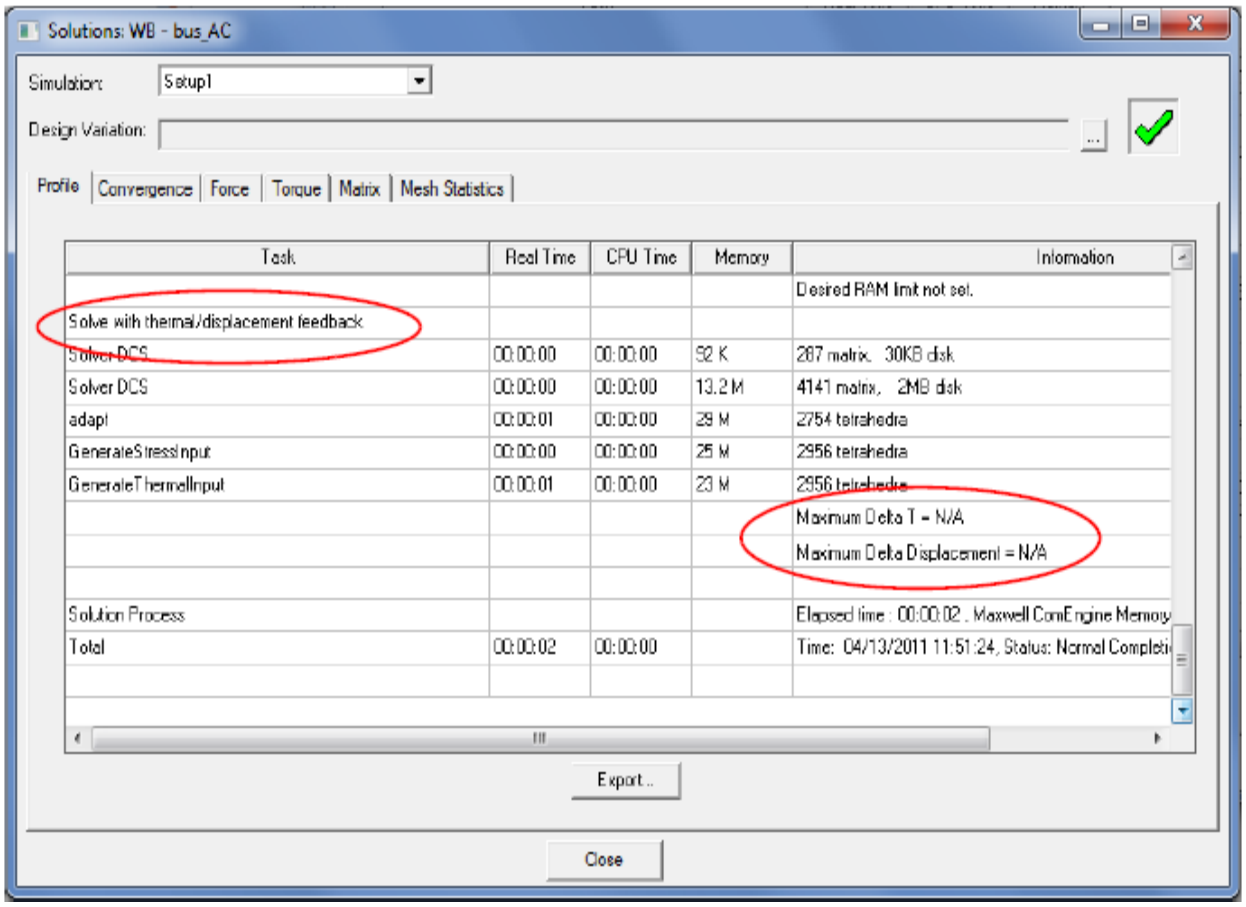
Example 1: “One” iteration

The following image illustrates a coupling setup where both the Thermal and Stress system are ready to be Updated. The Maxwell adaptive solution has converged in 3 passes. The Maxwell system has both thermal and stress feedback enabled.

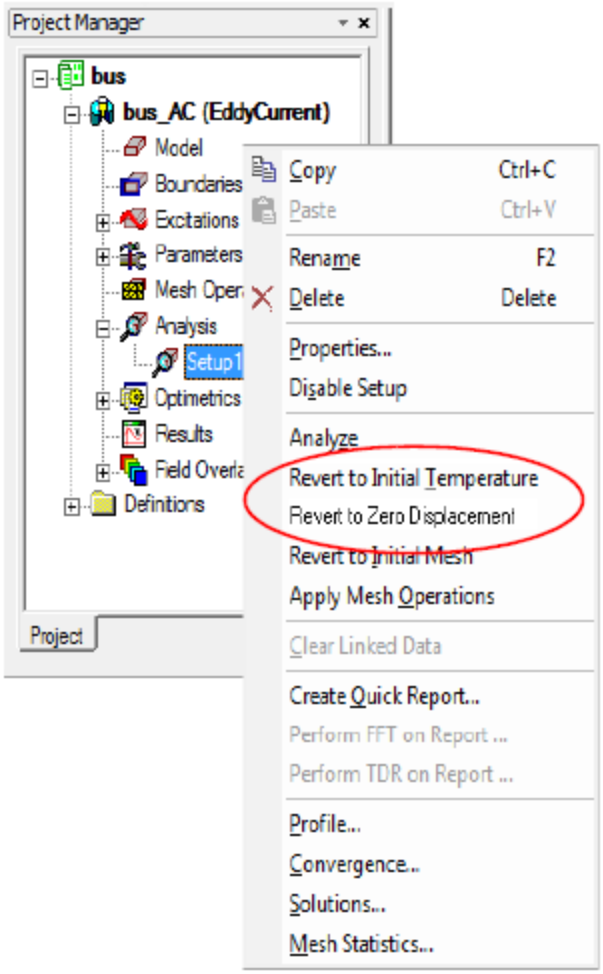


1. Right click on the **Solution** cell of Stress (Static Structural) and select **Update**.
   a. The **Setup** of Thermal will be Updated with “em” loss from Maxwell.
   b. The **Solution** of Thermal will be Updated and temperature will be exported to Maxwell.
   c. The **Setup** of Stress will be Updated with thermal force from Thermal and force density from Maxwell.
   d. Displacement will be exported to Maxwell after Stress finishes simulation.
2. Right click on the **Solution** cell of Maxwell and select **Enable Update**.
3. Right click on the **Solution** cell of Maxwell and select **Update**.
   a. Maxwell re-simulates the 3rd pass with its mesh and the temperature/displacement feedback.

b. The profile will show information about the feedback

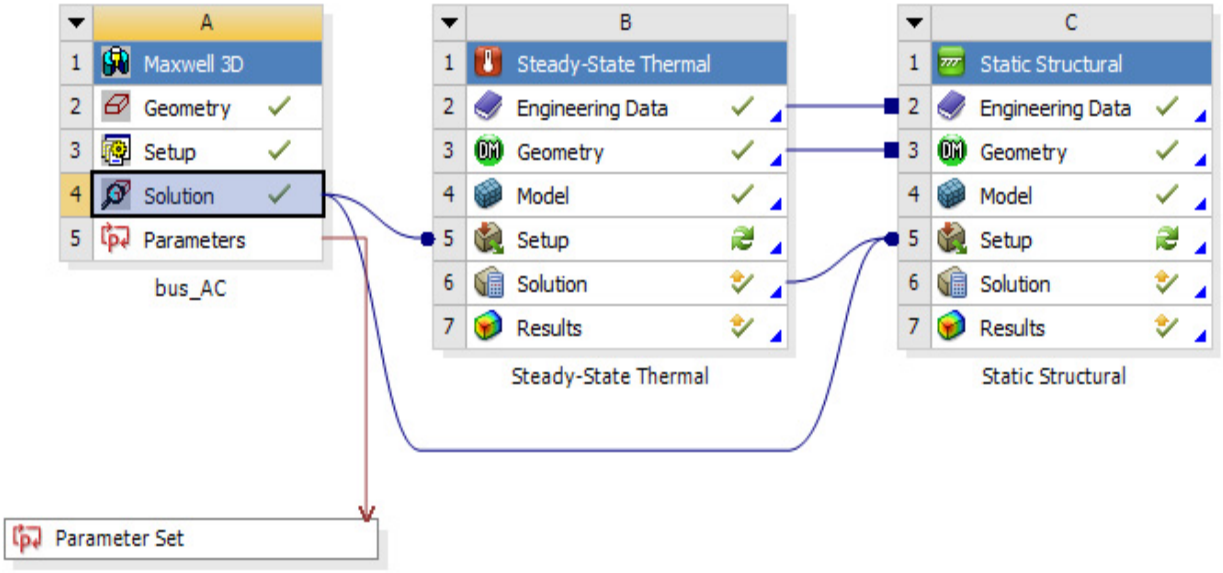


c. Both Revert to Initial Temperature and Revert to Zero Displacement menus are present in the analysis setup context menu.



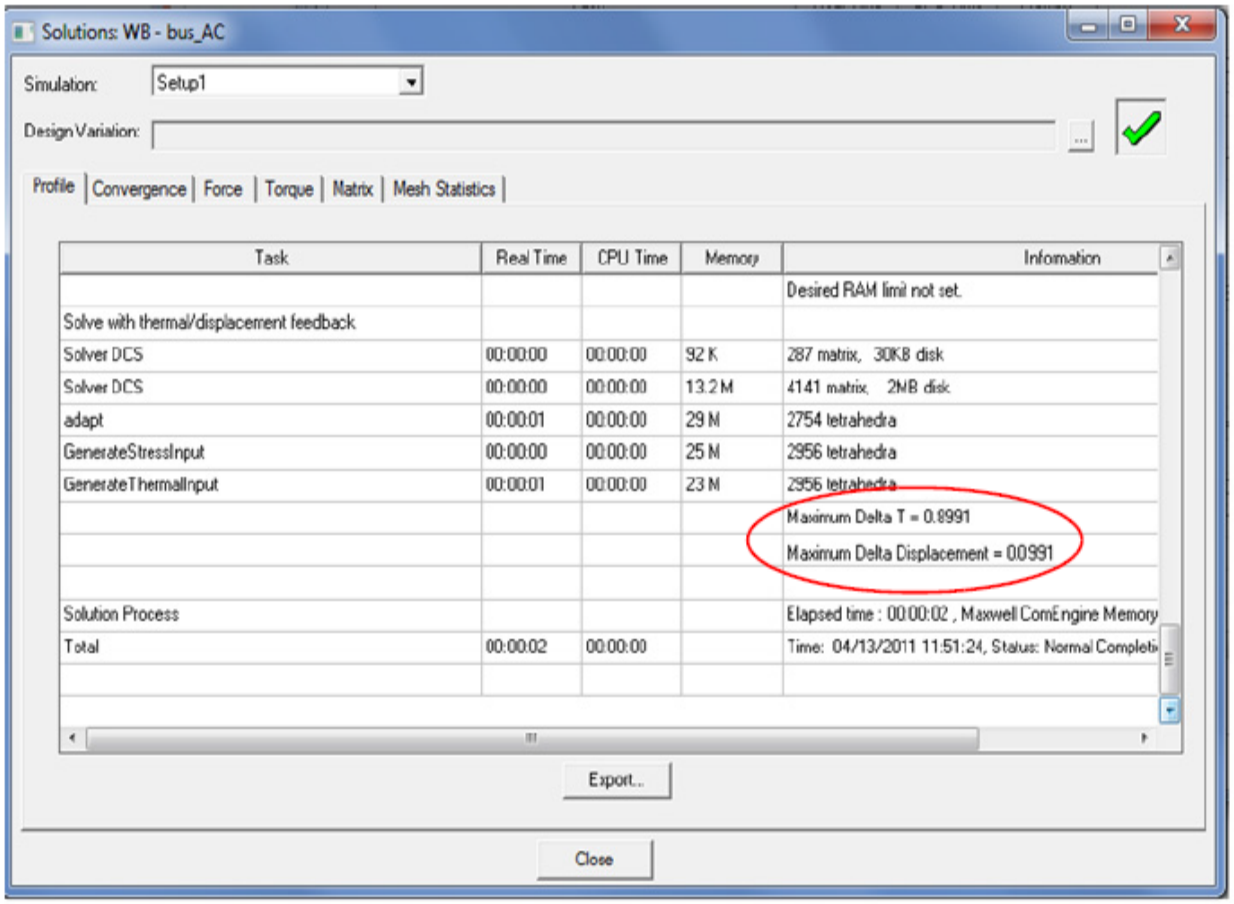
Example 2: Manual iteration

The following image illustrates the Workbench schematic after Example 1, where the 1st iteration has completed.



1. Right click on the **Solution** cell of Stress and select **Update**.
   a. The **Setup** of Thermal will be Refreshed and Updated with “em” loss from Maxwell.
   b. The **Solution** of Thermal will be Updated and temperature will be exported to Maxwell.
   c. The **Setup** of Stress will be Refreshed and Updated with thermal force from Thermal and force density from Maxwell.
   d. Displacement will be exported to Maxwell after Stress finishes simulation.
2. Right-click on the **Solution** cell of Maxwell and select **Enable Update**.
3. Right-click on the **Solution** cell of Maxwell and select **Update**.
   a. Maxwell re-simulates the 3rd pass with its mesh and *new* temperature/displacement feedback.

b. The profile will show information about the feedback.

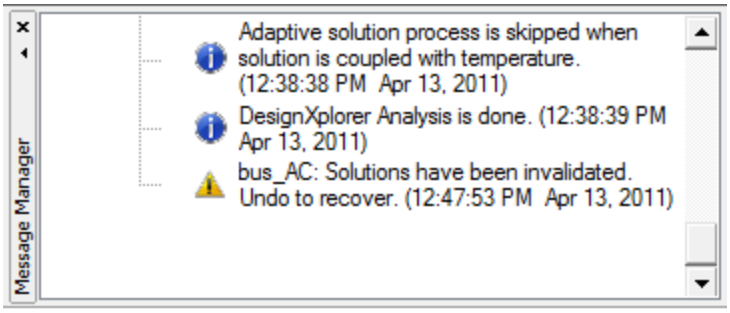


Note that the delta temperature and displacement is being reported in the profile.

Example 3: Revert Maxwell Solution

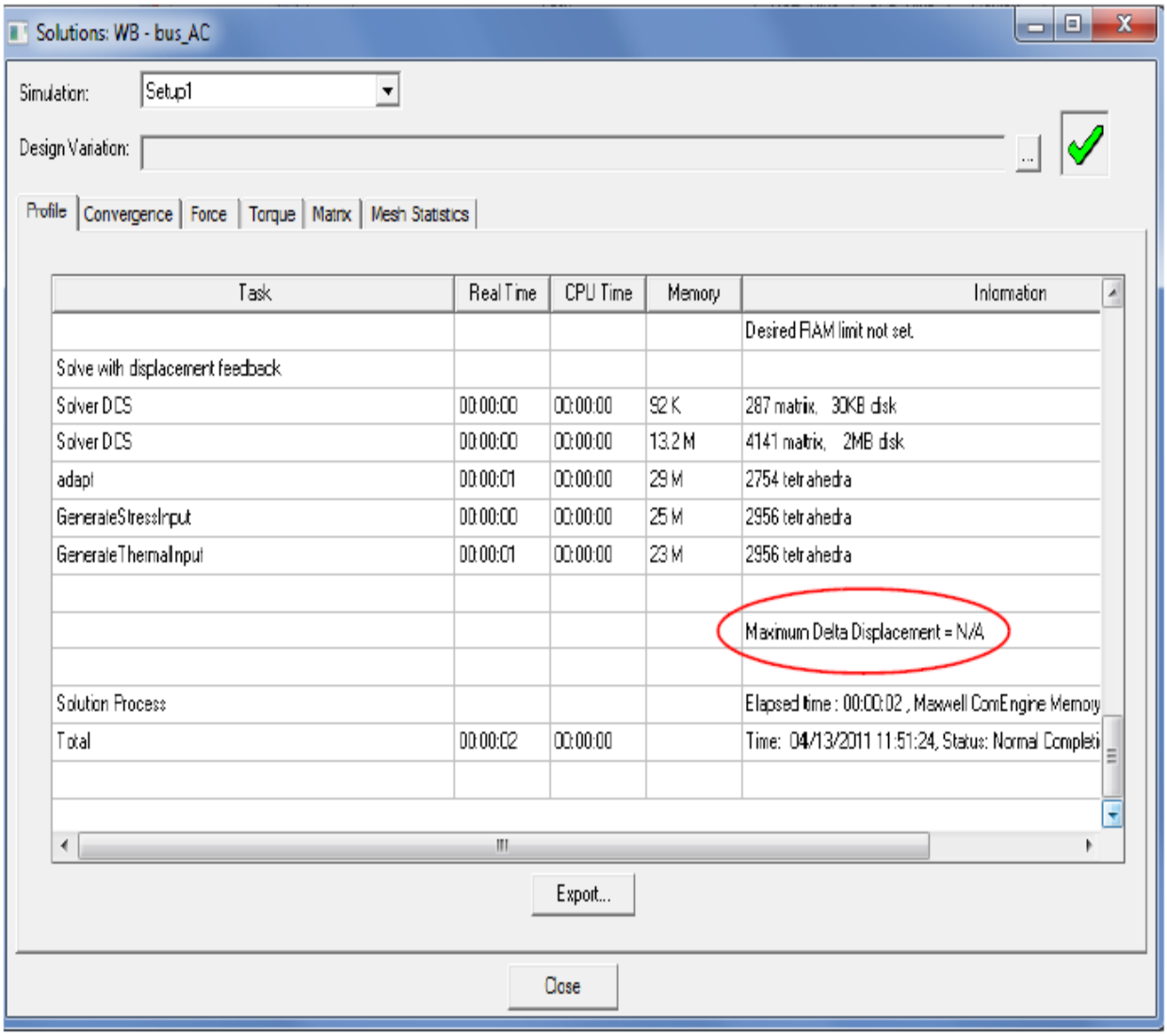
The temperature and displacement in the Maxwell solution can be reverted separately.

1. Users select **Revert to Initial Temperature** in Maxwell.
   A warning message displays notifying users about the invalidation of solution.



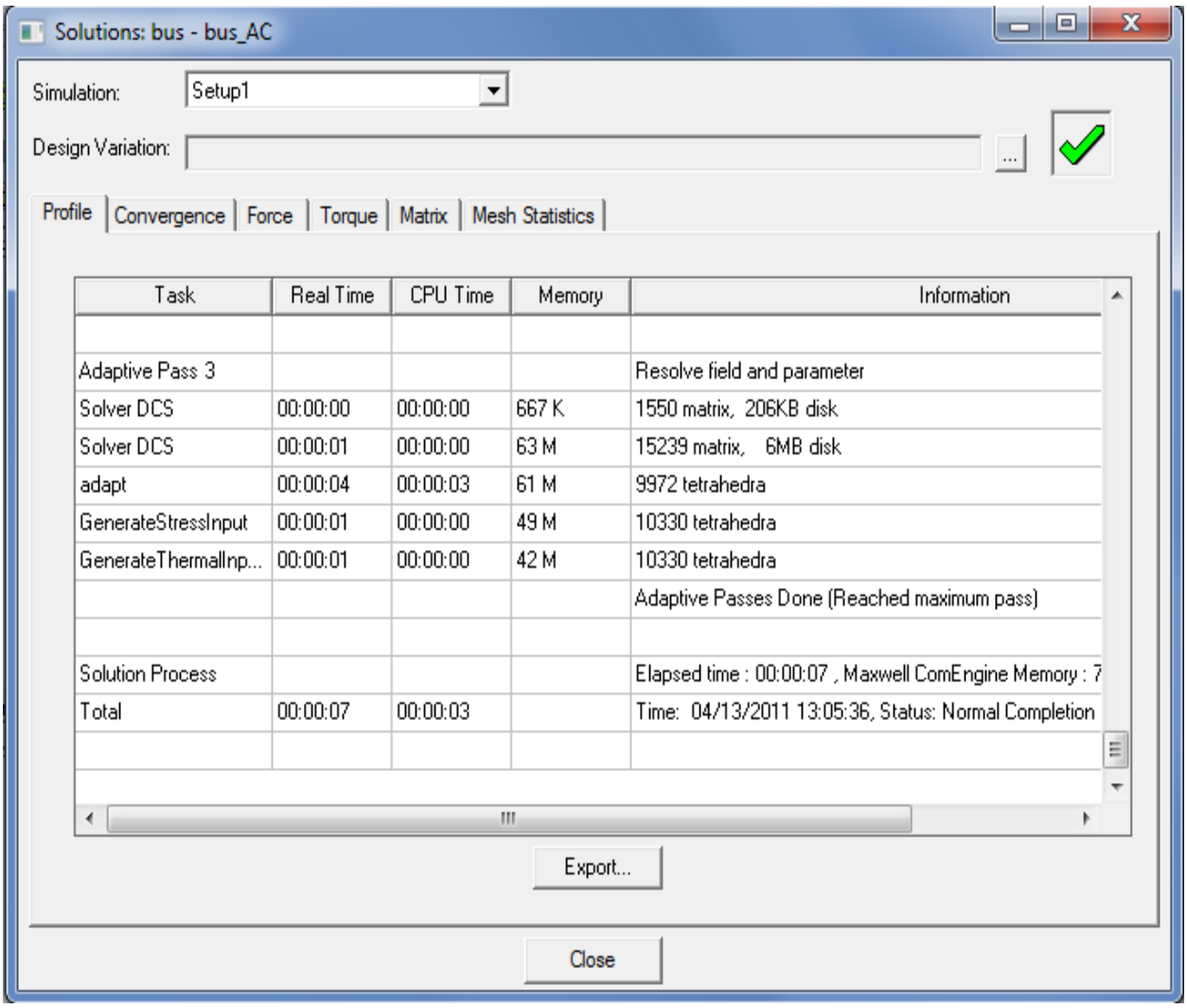
2. Right-click on Maxwell analysis setup and select **Analyze**.

   Maxwell re-simulates the 3rd pass with its mesh and displacement that was previously exported from Mechanical, but without temperature.



3. Users select **Revert to Zero Displacement** in Maxwell, followed by right-clicking on Maxwell analysis setup and selecting **Analyze**.

   Maxwell re-simulates the 3rd pass with its mesh without either temperature nor displacement.

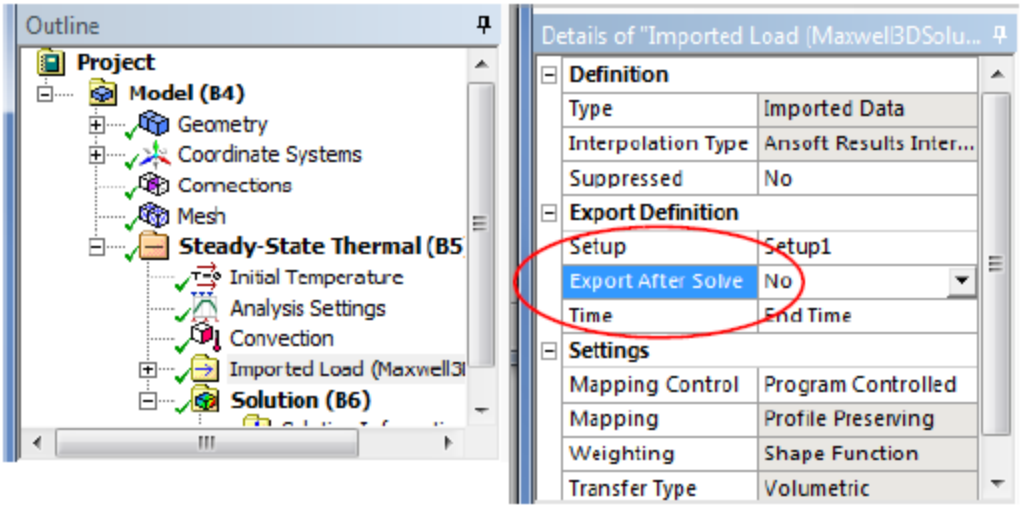


Example 4: Only Stress Feedback

The coupling setup is the same as in Example 1, with either one of the following differences:

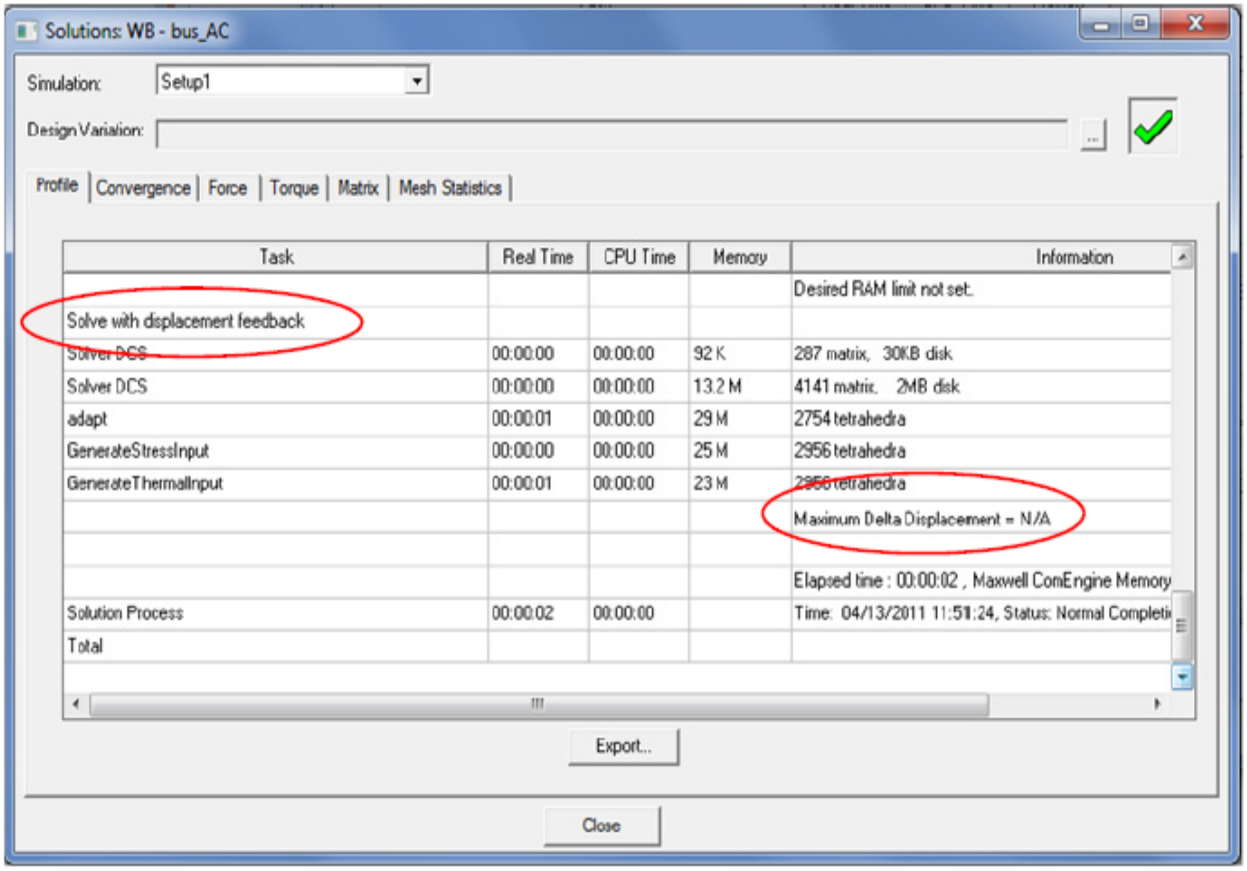
- The Maxwell system is not enabled for thermal feedback (via SetObjectDisplacement). Note that since Maxwell is not enabled to support feedback, the “Export Result” properties should be available in the thermal system.
- The Maxwell system is enabled for thermal feedback, but users choose not to “Export

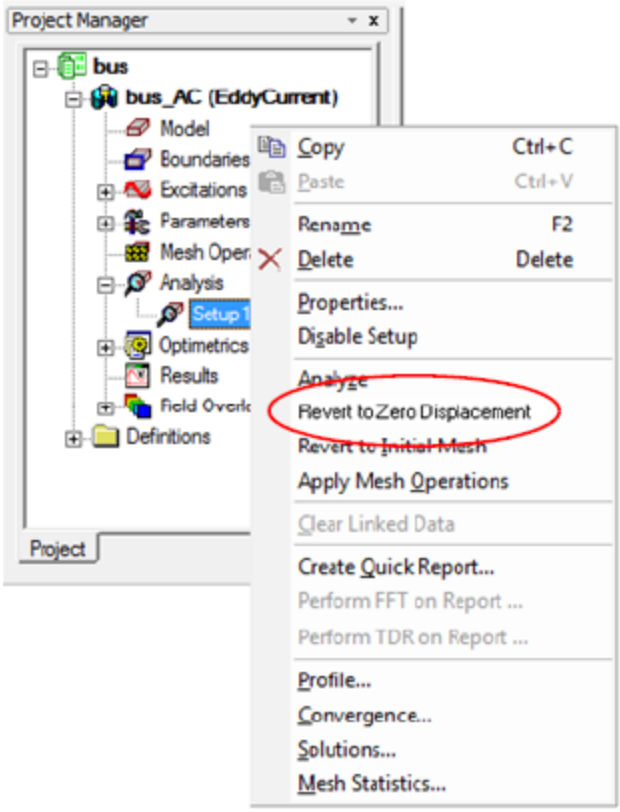
Result' in the thermal system. This is to disable the automatic "Export Results".



1. Right click on the **Solution** cell of Stress and select **Update**.
   a. The **Setup** of Thermal will be Updated with "em" loss from Maxwell.
   b. The **Solution** of Thermal will be Updated.
   c. The **Setup** of Stress will be Updated with thermal force from Thermal and force density from Maxwell.
   d. Displacement will be exported to Maxwell after Stress finishes simulation.
2. Right-click on the **Solution** cell of Maxwell and select **Enable Update**.
3. Right-click on the **Solution** cell of Maxwell and select **Update**.
   a. Maxwell re-simulates the 3rd pass with its mesh and the displacement feedback.
   b. The profile shows information about the feedback and **Revert to Zero Displacement**

will be available in the solve setup context menu.





System Coupling Analyses Using Maxwell

System Coupling facilitates and manages the execution of coupled simulations between multiple physics solvers, coupling active co-simulation participants using either System Coupling's graphical user interface (GUI) or its command-line interface (CLI).

Maxwell can engage as a coupling participant in thermal co-simulations with Ansys Fluent and Ansys Mechanical. System Coupling provides enhanced control over coupled simulation processes, including automatic starts and restarts for participants, the ability to manipulate the System Coupling data model, an interactive solution workflow, and multiple output formats for reviewing and postprocessing results.

In System Coupling analyses, each participant completes its physics and coupling setups in its own user interface generating a Participant Setup (.scp) file along with its solver file(s). The coupled analysis itself includes loading participants, setting values for coupling-related settings, specifying coupled analysis settings, and starting the run. Once the solution has started, System Coupling directs the execution of the coupled analysis – including general coupling management, participant communications, and transfers of data between participants.

Note	For detailed information on System Coupling in general, please refer to the *System Coupling User's Guide*. For details on working in System Coupling's GUI or CLI, see the Using System Coupling's User Interfaces section.

	For examples of Maxwell participating in co-simulations, review the coil-and-core induction heating, bar-and-coil induction heating, bus bar, and electric motor examples in *System Coupling Tutorials*.

Supported Capabilities and Limitations

The following capabilities are supported for System Coupling analyses with Maxwell:

- Multi-region data transfers for Maxwell participants.
- Maxwell 3D Eddy Current, Maxwell 3D Transient, and Maxwell 2D Transient Cartesian, XY geometry mode solutions.
- Update frequency controls for a Maxwell participant running a steady solution.
- Automatic and manual distribution parameters for parallel execution
- Dynamic remeshing and remapping during the execution of Maxwell 3D Eddy Current simulations with time-dependent motion or excitation defined. For details, see Mapping Process in the System Coupling User's Guide.
- Variable displacement for simulations with time-dependent motion defined.
- Thermal data transfers on bodies. EM Loss can be provided to other coupling participants, and temperature can be received from them.
- Motion data transfers on coupled bodies.
- Parallel/distributed processing.

The following are known limitations when using System Coupling with Maxwell:

- Only Maxwell 3D Eddy Current, Maxwell 3D Transient, and Maxwell 2D Transient, Cartesian, XY geometry mode solution types are supported.
- Maxwell 3D Eddy Current solution type is limited to a single frequency.
- Variable velocity is not supported for designs with time-dependent motion and/or excitations.
- Update frequency controls are not supported when Maxwell runs a solution with time-dependencies.
- Restarts from intermediate results are not supported.
- Restarts of transient analyses are not supported. When running an analysis of this type, Maxwell's Restarts Supported setting must be set to False, either in its .scp file or the System Coupling data model.

Using HPC Distribution Settings for Coupled Analyses

System Coupling supports the use of Maxwell's automatic and manual HPC distribution parameters for distributed coupled analyses. Each of these **Execution Control** settings corresponds to one of Maxwell's distributed analysis parameters.

- **Auto Distribution Settings**: Whether job distribution is performed automatically.

When auto-distribution is disabled, the following manual settings are available:

- **Include HPC Distribution Types**: List of the HPC distribution types to be included for distributed runs.
- **Number of Cores Per Task**: Number of HPC cores to be used per task.
- **Batch Options**: Batch option arguments to be applied to the run.

For more information, see *Using AEDT HPC Distribution Options* in the *System Coupling User's Guide*.

Variables Available for System Coupling

The following variables will be available on convective boundaries:

Name / Internal Name	Transfer Direction	Data Type	Physical Type
Loss / Loss	Output	Scalar	Heat Rate
Temperature / Temperature	Input	Scalar	Temperature

System Coupling Related Settings

System Settings

To enable coupling with Maxwell, the environment variable **ANSYSEM_ROOT231** must be set to the location of your Ansys Electromagnetics installation.

> **Note:** The environment variable name is ANSYSEM_ROOT<*major*><*minor*>, where <*major*> is the two digit major version number (the last two digits of the year); and <*minor*> is the one digit minor version number. For example, for the Ansys Electronics 2023 R1 release, the environment variable name is ANSYSEM_ROOT231. For the Student Version of the software, the variable name is ANSYSEM_ ROOTSV<*major*><*minor*>.

- For Windows, this environment variable is automatically set during the installation of the Ansys Electromagnetics Suite (for example, **C:\Program Files\AnsysEM\v231\Win64**).
- For Linux, you must set this environment variable manually (for example, **/opt/AnsysEM/v231/Linux64**).

> **Note:** The environment variable must be set on all hosts if the analysis is distributed to multiple hosts. For Windows, this environment variable must generally be set as a System environment variable (not a user environment variable) for distributed analysis so that it is set for the Ansoft RSM Service. For Linux, the user must ensure that the environment variable is set for the AnsoftRSM Service daemon processes.

Maxwell Settings

Solution Type:

Must be set to one of the following:

- **Eddy Current** for Maxwell 3D designs only.
- **Transient** for either Maxwell 3D or Maxwell 2D designs. For Maxwell 2D designs, you must also specify the **Cartesian, XY** geometry mode when selecting this solution type.

Materials:

Bodies to receive temperature data must have temperature-dependent properties. Note that when the same materials are used in multiple participants, common properties (e.g., conductivity) must be consistently defined.

Motion:

Motion must be appropriately and consistently defined for all bodies involved in the analysis.

Stationary bodies must not move from the reference position or participate in multiple reference frames during the co-simulation.

Motion should be defined with a (rigid body) displacement.

For more information, see Additional Participant Setup Considerations in the *System Coupling User's Guide*.

Creating a System Coupling Setup

The **System Coupling Setup** enables the exchange of data between participants by creating an interface between regions on the participant models.

To create a coupling setup, select (**Optimetrics>Add>System Coupling Setup**), and do one of the following:

- For Eddy Current designs, select the desired solution **Setup**, **Frequency**, and input/output quantities. For the **Temperature** quantity, click the **Object temperature** button to open the **Temperature of Objects** dialog in which you can choose to **Include Temperature Dependence**, and to set the temperatures for model objects composed of materials having temperature-dependent properties.
- For Eddy Current designs with time-dependent motion, create a dataset to define the position of the moving body, and design variables for the coupling time as provided by System Coupling and the displacement vector of the body.
- For Transient designs, select the desired solution **Setup**, **Start time**, **End time**, and input/output quantities. For the **Temperature** quantity, click the **Object temperature** button to open the **Temperature of Objects** dialog in which you can choose to **Include Temperature Dependence**, and to set the temperatures for model objects composed of materials having temperature-dependent properties.

- For Transient designs with time-dependent motion/excitation:
  - Define motion on coupled bodies.

Note	A profile or expression can be used to define rigid body motion using total displacement relative to the original object location.

  - Create design variables for both the coupling time (as provided by System Coupling) and the displacement vector of the body.
  - For Transient designs, Ansys Electromagnetics time is associated with System Coupling time. Maxwell uses this time to calculate excitation and position, but itself always begins its calculations at time zero.
  - Optionally, the following inputs can also be defined as a function of System Coupling time: **Maxwell Time Step Size**, **Loss Averaging Start Time**, **Loss Averaging Stop Time/Duration**.

Generating a System Coupling Participant Setup File

System Coupling uses each participant's Participant Setup (.scp) file and the corresponding Python (.py) configuration file to load the participant and its information into the data model.

To use System Coupling, you must generate a System Coupling participant file (**.scp**). Once the Maxwell setup is complete, generate the file using either of the following options:

- In the **System Coupling Setup** dialog box, the system coupling configuration files are generated when you click the **OK** button.
- In the **Project Manager** under **Optimetrics**, right-click your System Coupling Setup and select **Generate Configuration Files**.

In both cases, the configuration files needed by System Coupling (an **.scp** file and a **.py** file) are generated at the same time and in the same location as the Maxwell project file.

Licensing Considerations when using System Coupling

A distinct license is required for each coupling participant product, but no additional licenses are required for the System Coupling infrastructure itself.

Note, however, that Ansys EnSight is required for the postprocessing of System Coupling's interface results. If you are using a Multiphysics Licensing Bundle that does not include an EnSight license (e.g., the Ansys Mechanical Maxwell bundle), you will need a separate EnSight license to visualize co-simulation results.

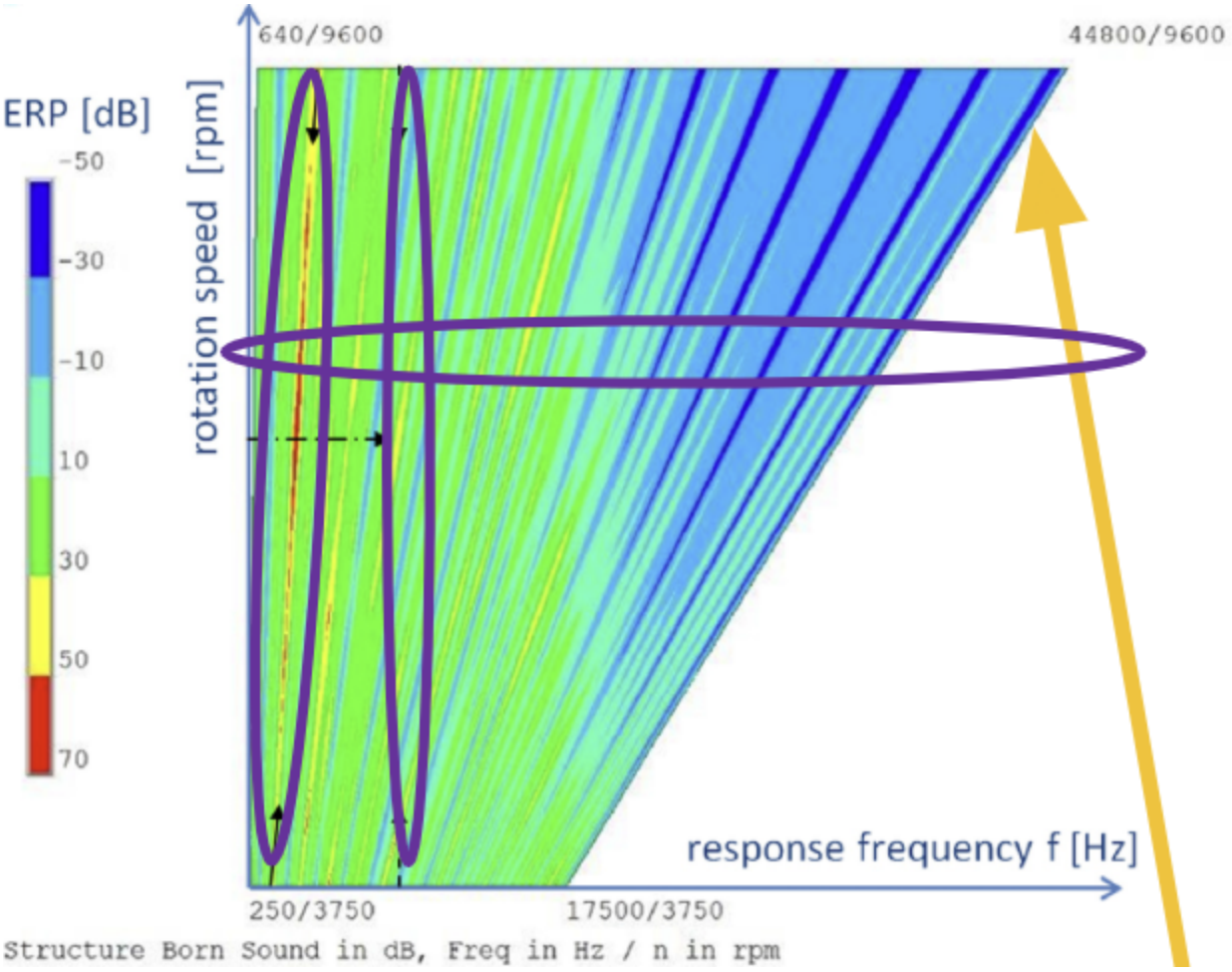
For more information on applicable licenses, see **Product Licensing Considerations** in the *System Coupling User's Guide*.

Maxwell Multiple RPM Harmonic Force Coupling

Waterfall analysis is a tool that is often used to evaluate the acoustic and/or vibratory performance of rotating machinery. Maxwell multiple RPM coupled with Ansys Harmonic Response provides a powerful tool to generate acoustic performance map of electric machines. The process involves calculating harmonic force using discrete Fourier transform (DFT) methods at each rpm

(revolutions per minute), and performing harmonic response analysis at every selected frequency with calculated harmonic force loads.

Waterfall computation is given by ERP=ERP(f, rpm) making use of load interpolation between computed RPM.



The Waterfall contour map showing the full acoustic fingerprint of the machine, where:

- Vertical Line: Given by one fixed eigenfrequency excited at different speeds of rotation.
- Horizontal Line: Single rpm load case yielding one frequency response function.
- Inclined Line: Excitation order of the electromagnetic force-density revealing a large ERP zone with its origin in single eigenfrequency that is met by both frequency and spatial load pattern of excitation and system mode.

To setup a Maxwell design for multiple RPMs coupling:

1. In the **Mechanical** tab of the **Motion Setup** dialog box, specify angular velocity (RPM) with a design variable.
2. In the **General** tab of the **Solve Setup** dialog box, specify the appropriate stop time and time step using the same design variable.

3. Add a parametric setup to sweep the variable.
4. Select the parametric setup as **Embedded Parametric Analysis** in the DesignXplorer setup.

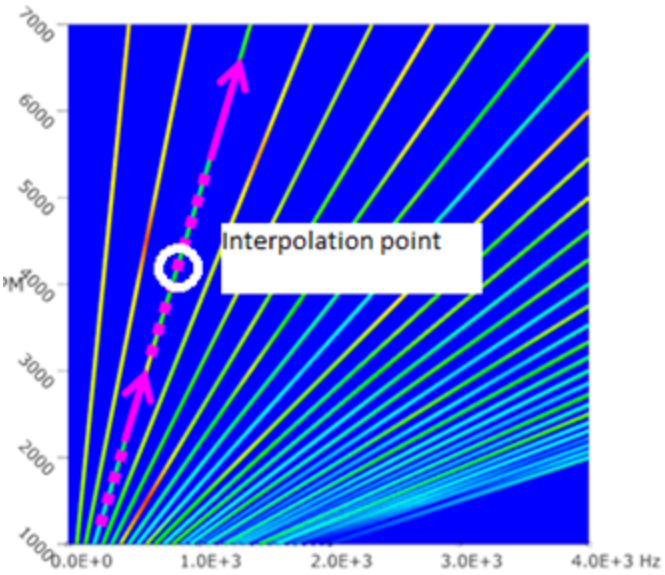
Related Topics

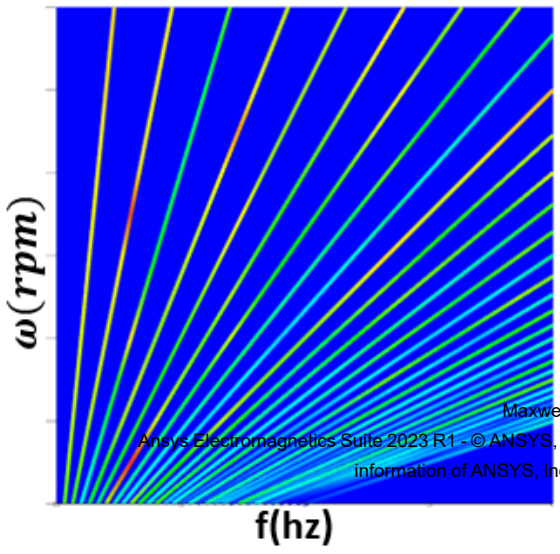
Harmonic Force Interpolation in Frequency-Speed Plane

Harmonic Force Interpolation in Frequency-Speed Plane

In order to generate the ERP waterfall diagram of electrical machines in Ansys Mechanical Harmonic Response, harmonic force distribution at different rotation speeds from Maxwell solver is provided as excitations of Mechanical. In general, the quality of the ERP diagram depends on the number of speed points that are solved. To reduce the workload in Maxwell, an option is provided to interpolate harmonic force in the frequency-speed plane with a limited number of solved speed points.

In electrical machines, harmonic forces only exist at certain frequencies (harmonics) at a given speed (as shown below):





Force components and corresponding frequencies at any given speed can be obtained by interpolation along those lines of harmonics:

$$F_k = F_{i-1,\ k} + \frac{(\omega - \omega_{i-1})}{\omega_i - \omega_{i-1}}\left(F_{i,\ k} - F_{i-1,\ k}\right)$$

$$f_k = f_{i-1,\ k} + \frac{(\omega - \omega_{i-1})}{\omega_i - \omega_{i-1}}\left(f_{i,\ k} - f_{i-1,\ k}\right)$$

$$i = 2,\ \cdots\cdots M$$

where:

F : force vector

f : frequency

ω : speed of rotation

k : index of harmonics

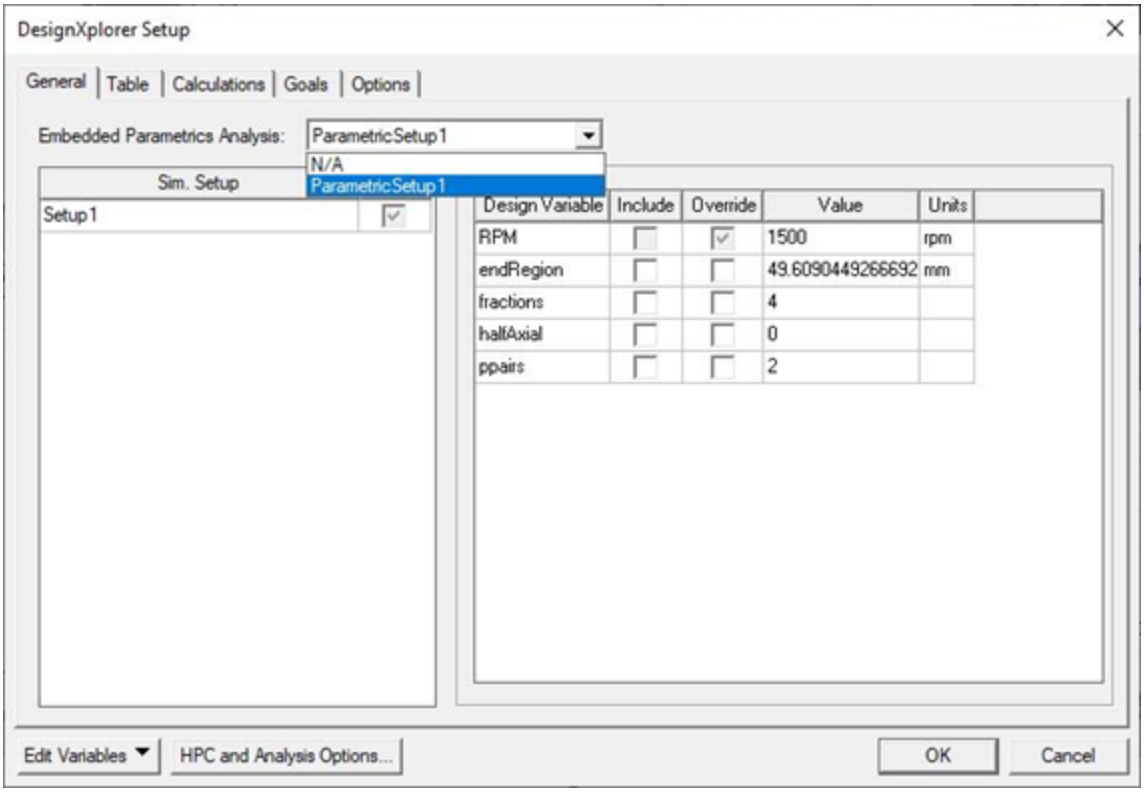
i : index of solved speed points

M : total number of solved speed points

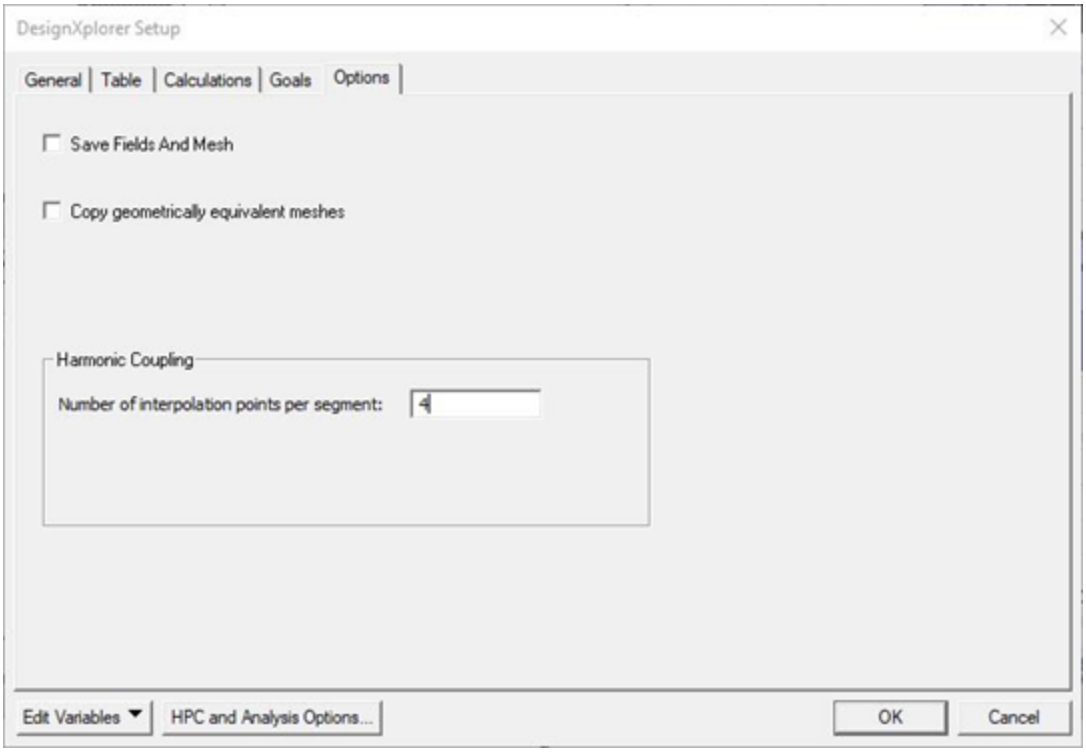
Note	To use this feature you must open Maxwell from within Workbench. For detailed information on setting up and using harmonic force coupling in Workbench, refer to the Workbench help topic: *Importing Data into a Harmonic Analysis*.

To setup the number of interpolation points per segment:

1. Open the **DesignXplorer Setup** panel and set Parametric Sweep (RPM sweep) as the **Embedded Parametric Analysis**.



2. On the Options tab, enter the **Number of interpolation points per segment**. (This value must be a non-negative integer.)



Related Topics

Maxwell Multiple RPM Harmonic Force Coupling

Maxwell Harmonic Force Calculations and Coupling

- For 2D/3D *transient* and *eddy current* designs Maxwell supports *object-based* one-way harmonic force coupling via Ansys Workbench . Maxwell enables you to select multiple objects for force calculations for harmonic coupling with Workbench projects.
- For 2D/3D *eddy current* designs, Maxwell provides *element-based* solutions for one-way harmonic force coupling via Ansys Workbench. For eddy current element-based solutions, there is no need to select objects in Maxwell, because the harmonic surface force is a default output in the solver.
- For 2D/3D *transient* designs, Maxwell provides *element-based* solutions for one-way harmonic force coupling via Ansys Workbench. Element-based harmonic force calculations can also be enabled for stand-alone transient designs.
- Maxwell supports one-way coupling with various third-party software for transient designs.
- Maxwell supports one-way coupling with Ansys motion for transient designs.

Related Topics

"Maxwell to Ansys Object-based Harmonic Force Coupling" on the next page

"Maxwell to Ansys Element-based Harmonic Force Coupling" on page 17-51
"Export Transient/Harmonic Force from Maxwell " on page 17-57
"Maxwell One-way Coupling with Ansys Motion " on page 17-65
Force Densities in Maxwell
Force Density Calculation in Maxwell

Maxwell to Ansys Object-based Harmonic Force Coupling

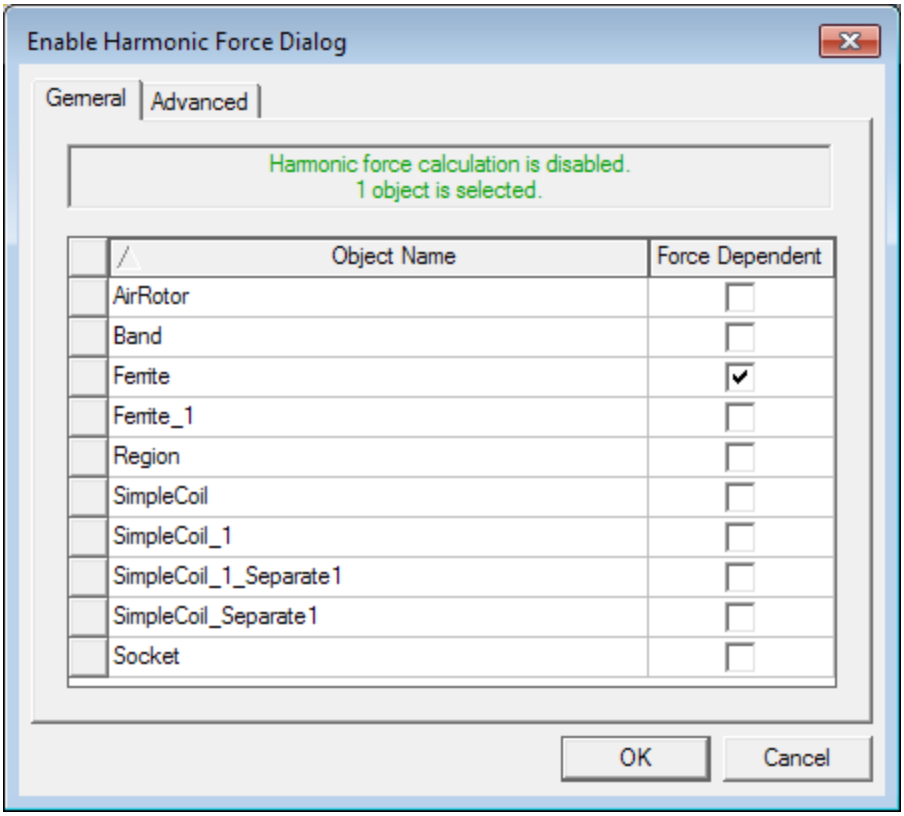
Maxwell 2D/3D supports object-based one-way harmonic force coupling for *transient* and *Eddy current* designs via the Workbench schematic. Maxwell enables the user to select multiple objects for force calculations for harmonic coupling with Workbench. You can also setup multiple RPM Harmonic Force Coupling.

> **Note:** For object-based harmonic force calculation, a full-axial model is required

For detailed information on setting up and using harmonic force coupling in Workbench, refer to the Workbench help topics: *Importing Data into a Harmonic Analysis* and *Importing Data into a Thermal or Structural (Static or Transient) Analysis*.

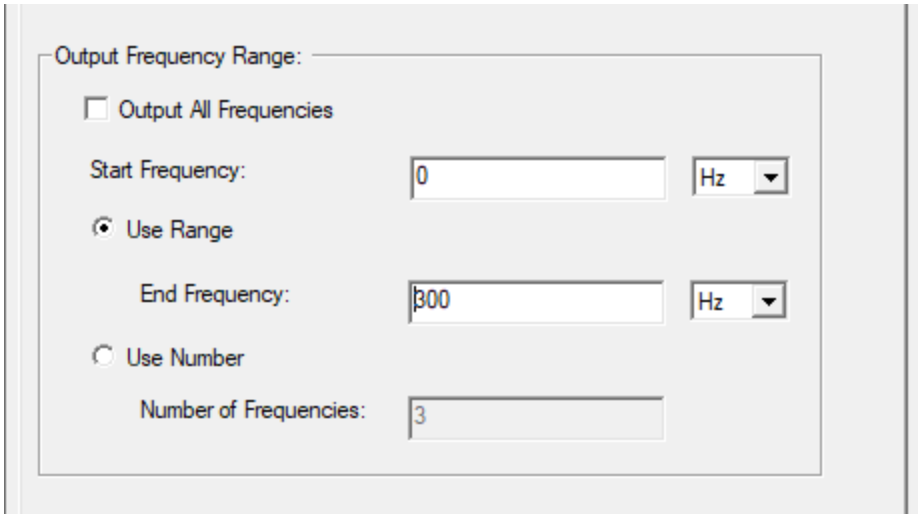
To enable object-based harmonic force calculations in Maxwell for use in Workbench:

1. Either select **Maxwell 2D** (or **Maxwell3D**)**>Enable Harmonic Force Calculation**, or right-click on the Design in the project tree and select **Enable Harmonic Force Calculation** to open the **Enable Harmonic Force Dialog** box. The dialog has two tabs: **General**, and **Advanced**.
2. Do one of the following:
    - For *Eddy current* designs, on the **General** tab, check the **Force Dependent** objects for which force calculations are desired.

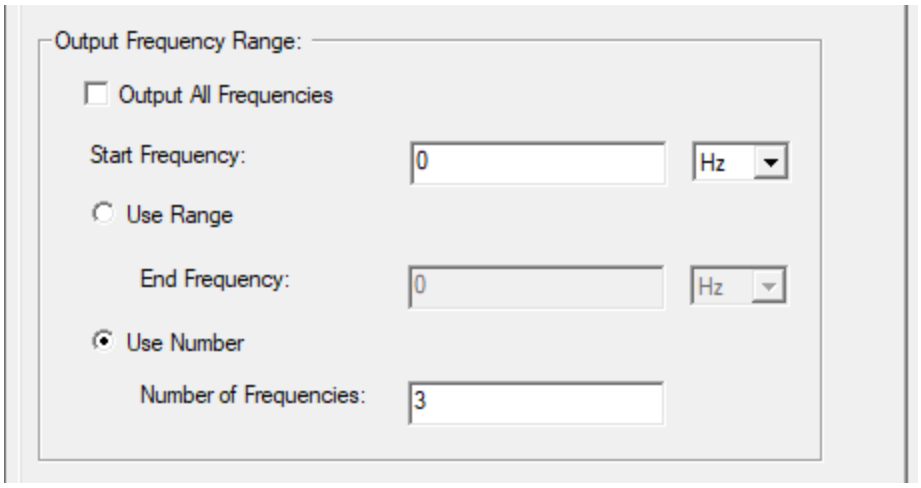


- For *transient* designs, on the **General** tab, select **Object Based** as the harmonic force **Type**. Select the **Harmonic Force** radio button. Then check the **Force Dependent** objects for which force calculations are desired.

3. The **Advanced** tab provides additional settings to allow you to control collection of data from the force calculations.
   - For *2D and 3D transient* designs (either with rotational motion or with non-rotational motion), the **Output Frequency Range** panel allows you to specify the frequency range of calculated harmonic force components. The default is to **Output All Frequencies** that are calculated. There are two options to set the range of output frequencies:
     - **Use Range** – specify the **Start Frequency** and **End Frequency**. Parameterization of **Start Frequency** and **End Frequency** is supported.



- **Use Number** – specify the **Start Frequency** and the **Number of Frequencies**. Parameterization of **Number of Frequencies** is supported.



- For *transient* designs that include rotational motion:

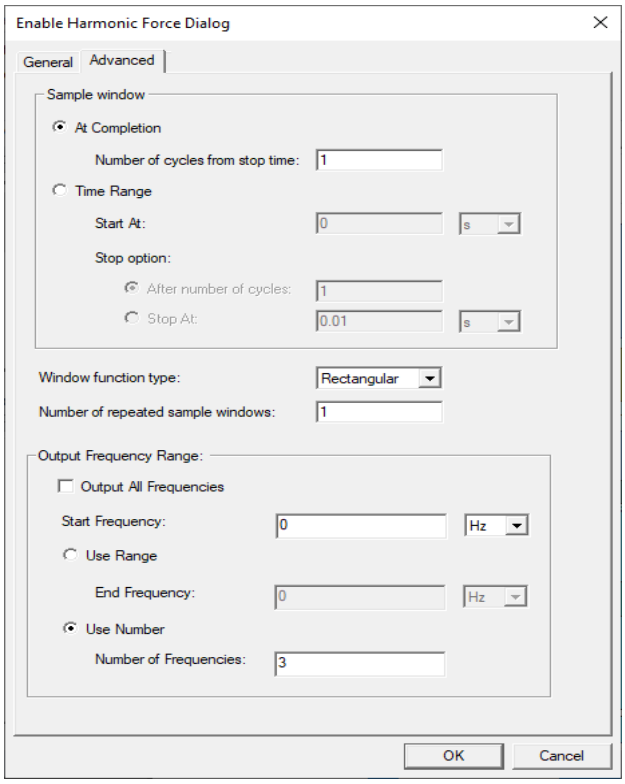
> **Note:** When using Partial Simulation for Full Rotational Machine Models, please select *all* teeth tips as objects for harmonic force calculation

  - The **Sample Window**'s collect samples **At completion** setting allows you to choose the **Number of cycles from stop time** for which data is collected. For example, if one cycle equals 1 second, and you solve for 3.5 seconds, setting the **Number of cycles from stop time** to 2 results in data being collected for the period from 1.5 seconds to 3.5 seconds. Alternatively, user can specify a **Time Range** to collect data, for which user can set a data collection **Start At** and a data collection **Stop option** – either stop data collection after a user-specified number

of cycles (**After number of cycles**) or at a user-specified **Stop At** time.

- The **Window function type** options setting is used for Discrete Fourier Transform (DFT) of harmonic forces. Please refer to *Window Function Options for DFT of Harmonic Forces* for details on this setting.
- In the frequency domain of harmonic force, the base frequency is determined by the range of time window of data collection ($f=1/\Delta T$). In order to get lower base frequencies, one must increase the time window of data collection, which means increasing the simulation time. It is waste of resources to do so when the solution reaches steady state/periodic in a transient simulation. Instead, you can set the **Number of repeated sample windows** to expand the range of time window of data collection without increasing simulation time by repeating the collected data *n* times before doing DFT. By applying this option, you can get a lower base frequency and higher frequency resolution, but the magnitudes of force components are close to zero at the introduced frequencies.

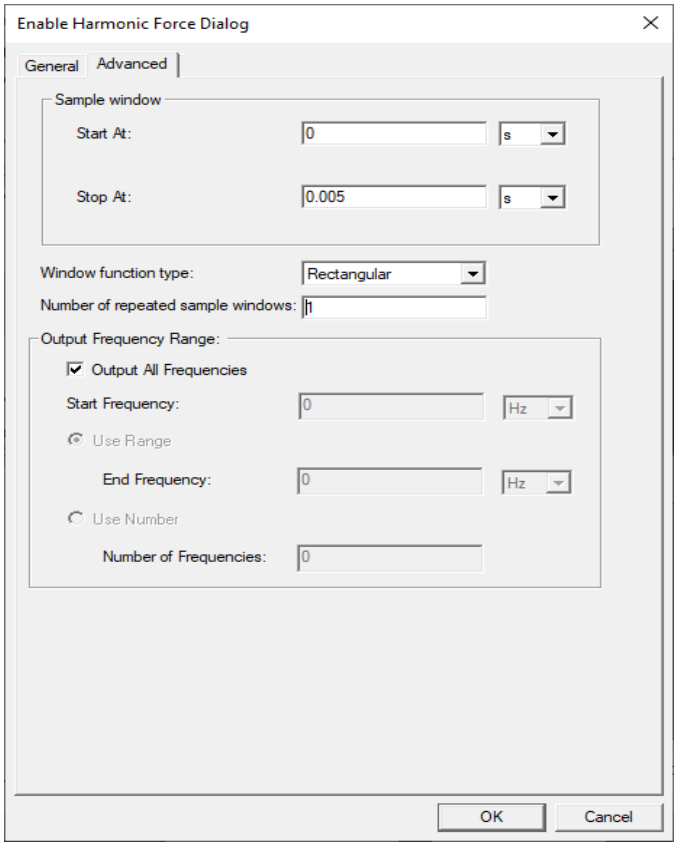
> **Note:** The **Number of repeated sample windows** setting is enabled only for the **Rectangular** (the default) **Window function type**.



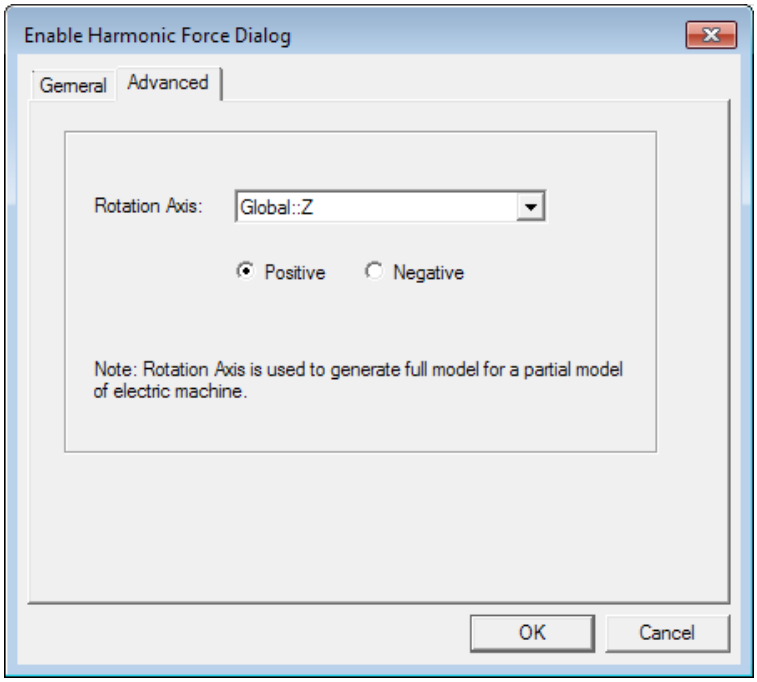
- For *transient* non-rotational motion designs:
  - The **Sample Window** of the **Advanced** tab allows you to set the data collection **Start At** and **Stop At** times.

- The **Window function type** options setting is used for Discrete Fourier Transform (DFT) of harmonic forces. Please refer to *Window Function Options for DFT of Harmonic Forces* for details on this setting.
- You can set the **Number of repeated sample windows** to expand the range of time window of data collection.

> **Note:** The **Number of repeated sample windows** setting is enabled only for the **Rectangular** (the default) **Window function type**.



- For *Eddy current* designs, the Advanced tab allows you to set the Rotation Axis and the axis direction so that the Maxwell solver can generate a full model if a partial model is used in the design. You can create relative coordinate systems to add more axes.



Harmonic Eddy force solutions are kept as part of Maxwell design solutions. If Workbench harmonic design settings are changed, the Maxwell solutions will be invalidated.

4. Click **OK** to close the dialog.

Related Topics

Maxwell to Ansys Element-based Harmonic Force Coupling

Force Densities in Maxwell

Force Density Calculation in Maxwell

Harmonic Force Calculation in Maxwell 2D with Skew Model

Transient Report of Object-based Harmonic Force Components

Maxwell One-way Coupling with Ansys Mechanical

Force Densities in Maxwell

Let $T_{\alpha\beta}$ be the Maxwell stress tensor (refer to reference [1] for detail of the Maxwell stress tensor). Then the force acting on an object can be calculated by:

$$F_\alpha = \sum_\beta \int_V \frac{\partial}{\partial x_\beta} T_{\alpha\beta} d^3x \qquad (1)$$

Application of the divergence theorem to volume integral gives:

$$F_\alpha = \oint_S \sum_\beta T_{\alpha\beta} n_\beta ds \quad (2)$$

Note that $\sum_\beta T_{\alpha\beta} n_\beta$ is the force per unit area transmitted across the surface. Namely, it is the surface force density, which is denoted as:

$$f_\alpha^s = \sum_\beta T_{\alpha\beta} n_\beta \quad (3)$$

Also, from (1), you can define a volume force density by stress tensor as:

$$f_\alpha^v = \sum_\beta \frac{\partial}{\partial x_\beta} T_{\alpha\beta} \quad (4)$$

For a magnetostatic field (similar forms for electric field), the explicit form of the Maxwell stress tensor is:

$$T_{\alpha\beta} = H_\alpha B_\beta - \frac{1}{2}\vec{B} \bullet \vec{H}\delta_{\alpha\beta} \quad (5)$$

So the surface force density can be calculated as:

$$f_\alpha^s = (\vec{B} \bullet \vec{n})\vec{H} - \frac{1}{2}\vec{B} \bullet \vec{H}\vec{n} \quad (6)$$

The volume force density can be calculated by (4) and using the virtual work principle (refer to reference [2] for details).

Reference [1] J.D. Jackson, Classical Electrodynamics, 3rd edition, Wiley, 1999.

Reference [2] A. Bossavit: "Eddy-currents and forces in deformable conductors", in Mechanical Modellings of New Electromagnetic Materials (Proc. IUTAM Symp., Stockholm, April 1990, R.K.T. Hsieh, ed.), Elsevier (Amsterdam), 1990, pp. 235-42.

Force Density Calculation in Maxwell

Maxwell solvers can be coupled with Ansys Mechanical either one-way or two-way. Maxwell solvers provide volumetric force density/surface force density as load of Mechanical solvers. Force density is calculated by Maxwell stress tensor.

Maxwell stress tensor for magnetic fields:

$$\bar{S} = \frac{1}{\mu_0}\begin{bmatrix} B_x^2 - B^2/2 & B_xB_y & B_xB_z \\ B_xB_y & B_y^2 - B^2/2 & B_yB_z \\ B_xB_z & B_yB_z & B_x^2 - B^2/2 \end{bmatrix}$$

Where $B^2 = B_x^2 + B_y^2 + B_z^2$

The force on a surface can be calculated as:

$$\vec{F} = \frac{1}{\mu_0} \iint dA\overline{S} \cdot \boldsymbol{n}$$

Where $\mathbf{n}$ is outward normal direction of the face.

The surface force density is calculated from:

$$\vec{f} = \frac{\vec{F}}{A}$$

Where ***A*** is the area of the surface.

The volumetric force density in an element is calculated from:

$$\vec{f} = \frac{1}{V}\sum_{i=0}^{N} \vec{F_i}$$

Where ***V*** is volume of the element, and ***N*** is the number of surfaces (sides in 2D) of this element.

Harmonic Force Calculation from Maxwell

Maxwell can calculate the harmonic force on each tooth of a rotating machine. This feature enables the one-way coupling between Maxwell and Harmonic Response of ANSYS. The force is object based, not element based. The harmonic momentum reference to the geometric center of the object is also provided at the same time. This feature is also extended to any individual object.

There are two options to calculate the force of an individual tooth:

a. Split the stator/rotor to form separate tooth object, and integrate forces on all out surfaces of this tooth except the splitting surfaces.
b. Draw an arc (2D) or a cylindrical face (3D) near the tooth in the air gap, and integrate forces on this face.

Discrete Fourier Transform (NDFT) with non-uniform time interval

In the general case, the definition of the Non-uniform Discrete Fourier Transform (NDFT) is the same as the definition with constant time interval, taking into consideration that the samples can be taken at irregular intervals in time (t_n). That is to say that the samples x(k) of the irregular Fourier transform are taken at multiples of a quantity $\Delta\omega$, which is a fixed quantity in the Fourier domain. The fixed quantity $\Delta\omega$ in the regular case corresponds to $2\pi/T$. The extension from regular to irregular sampling, therefore, depends on the duration of the signal *x(t)* and not on the fact that the samples are taken at regular or irregular intervals.

A finite sample measured at ***N*** points in a period of T:

$$x(k) = \begin{cases} 0, & k < 0 \\ F(t_k), & 0 \le k \le (N-1) \\ 0, & k \ge N \end{cases}$$

The definition of the nonuniform discrete Fourier transform (NDFT)is as follows:

$$X(k) = \frac{1}{N}\sum_{n=0}^{N-1} x(n) e^{-j(t_n - t_0)k\frac{2\pi}{T}}$$

The inverse NDFT is given by:

$$x(t_n) = \sum_{k=0}^{N-1} X(k) e^{j2\pi\frac{k(t_n - t_0)}{T}}$$

Data collection:

The solver records the components of force and momentum at all teeth at every time step.

$$F_x: F_x(0),\ F_x(\Delta t),\ F_x(2\Delta t),\ \cdots\cdots,\ F_x(K\Delta t)$$
$$F_y: F_y(0),\ F_y(\Delta t),\ F_y(2\Delta t),\ \cdots\cdots,\ F_y(K\Delta t)$$
$$F_z: F_z(0),\ F_z(\Delta t),\ F_z(2\Delta t),\ \cdots\cdots,\ F_z(K\Delta t)$$
$$M_x: M_x(0),\ M_x(\Delta t),\ M_x(2\Delta t),\ \cdots\cdots,\ M_x(K\Delta t)$$
$$M_y: M_y(0),\ M_y(\Delta t),\ M_y(2\Delta t),\ \cdots\cdots,\ M_y(K\Delta t)$$
$$M_z: M_z(0),\ M_z(\Delta t),\ M_z(2\Delta t),\ \cdots\cdots,\ M_z(K\Delta t)$$

If $t_{end} - t =< \frac{2\pi}{period * rotation\_speed}$, then keep records.

Generate harmonic force from partial model to full model

With symmetry-multiplier (period in 3D)>1, Maxwell solvers automatically generate harmonic force for the full model from the partial model

$$r^* = Ar$$

Where *r* is the vector to be rotated, *r** is the vector after rotation. The rotation matrix in 3D:

$$A = \begin{bmatrix} cos\theta + u_x^2(1-c0s\theta) & u_x u_y(1-c0s\theta) - u_z sin\theta & u_x u_z(1-c0s\theta) + u_y sin\theta \\ u_x u_y(1-c0s\theta) + u_z sin\theta & cos\theta + u_y^2(1-c0s\theta) & u_y u_z(1-c0s\theta) - u_x sin\theta \\ u_x u_z(1-c0s\theta) - u_y sin\theta & u_y u_z(1-c0s\theta) + u_x sin\theta & cos\theta + u_z^2(1-c0s\theta) \end{bmatrix}$$

The rotation matrix in 2D:

$$A = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

Vectors that need to be rotated: position (**r**), force (**F**) and moment (**M**)

Spatial and Temporal 2D Discrete Fourier Transform (DFT)

Electromagnetic force components on teeth inside an electric machine are spatial (in circumferential direction) and temporal (transient). The magnetic force density acting on the stator along the airgap is dependent on time and on space. This spatial and temporal force distribution can be expanded by means of Fourier analysis to calculate its time and space harmonic waves. This analysis can be used to improve NVH (noise, vibration, and harshness) performance.

The magnetic force density acting on the stator along the air gap is dependent on time and on space. This spatial and temporal force distribution can be expanded by means of Fourier analysis to calculate its time and space harmonic waves.

The force components on teeth can be expressed as function f(x, t), where x represents spatial, and t denotes temporal. The function f (x, t) is decomposed to a linear combination of harmonic (sines and cosines, more generally orthogonal) functions.

Direct Transformation

$$F(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) \exp\left[-2\pi i \left(\frac{mu}{M} + \frac{nv}{N}\right)\right]$$

Where:

u = 0, 1, …… M-1, wave number in spatial,

v = 0, 1, ……. N-1, wave number in temporal.

Window Function Options for DFT of Harmonic Forces

Window function options for Discrete Fourier Transform (DFT) of harmonic forces are available for Maxwell2D and 3D transient solvers and support both rotational and non-rotational models.

The Discrete Fourier Transform (DFT) operates on a finite-length time sequence to compute its spectrum. For a continuous signal like a sine wave, you need to capture a segment of the signal in order to perform the DFT. Usually, you also need to apply a window function to the captured signal before taking the DFT. A window function (also known as an apodization function or tapering function) is a mathematical function that is zero-valued outside of some chosen interval, normally symmetric around the middle of the interval, usually near a maximum in the middle, and usually tapering away from the middle. The mismatch in amplitude between the two ends of the signal distorts the spectrum, a phenomenon called spectral leakage. Spectral leakage can be caused by discontinuities in the non-integer number of periods in sampled signals that usually happen in asynchronous machines. This can be improved using windowing. Maxwell provides several window functions for the DFT of harmonic forces. These include:

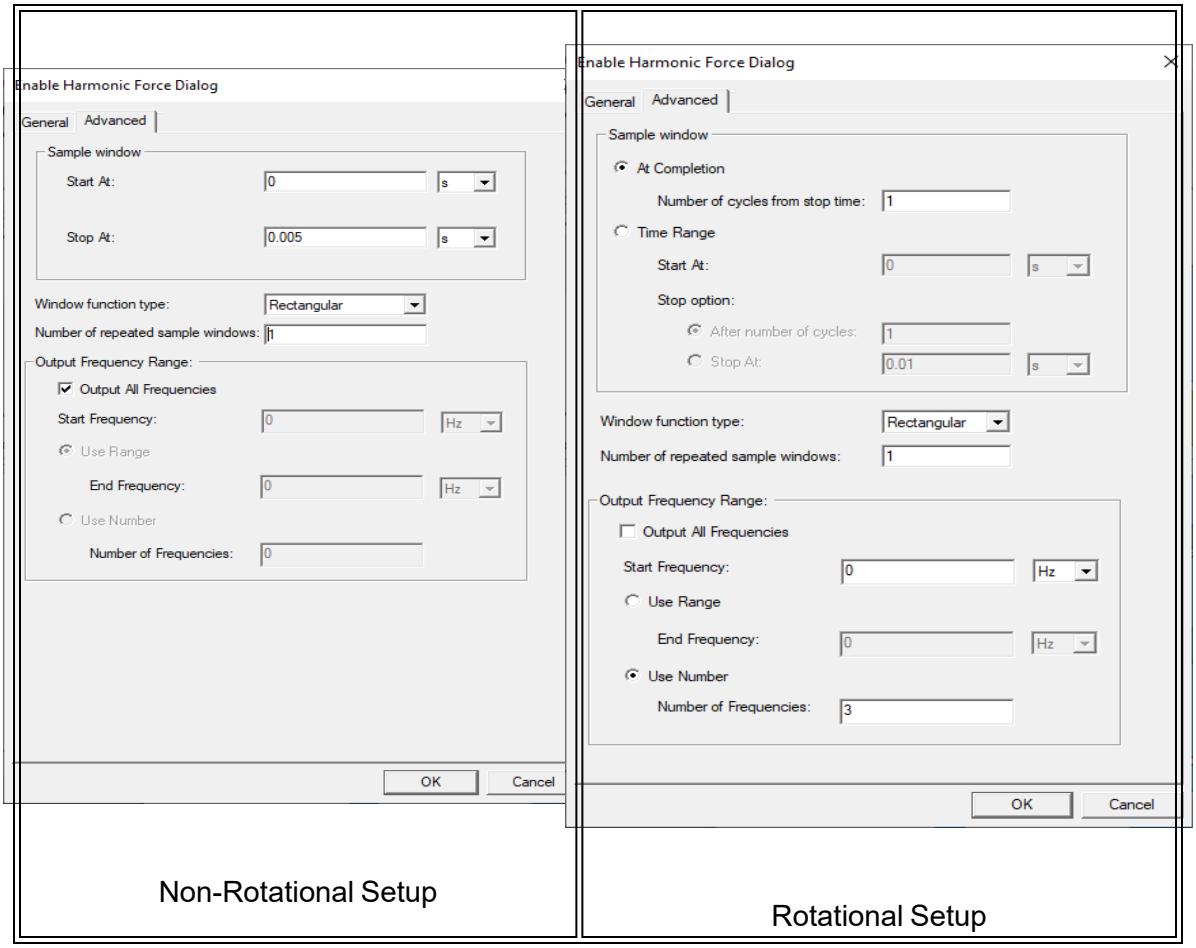
- Rectangular (default)
- Triangular
- Van Hann

- Hamming
- Blackman
- Lanczos
- Welch

Please refer to *FFT Window Functions* for detailed information on each of these window function types.

To select the various window function options:

1. Either select **Maxwell 2D** (or **Maxwell3D**)**>Enable Harmonic Force Calculation**, or right-click on the Design in the project tree and select **Enable Harmonic Force Calculation** to open the **Enable Harmonic Force Dialog** box.
2. Specify the **Window function type** using the drop-down list on the **Advanced** tab.



Harmonic Force Calculation in Maxwell 2D with Skew Model

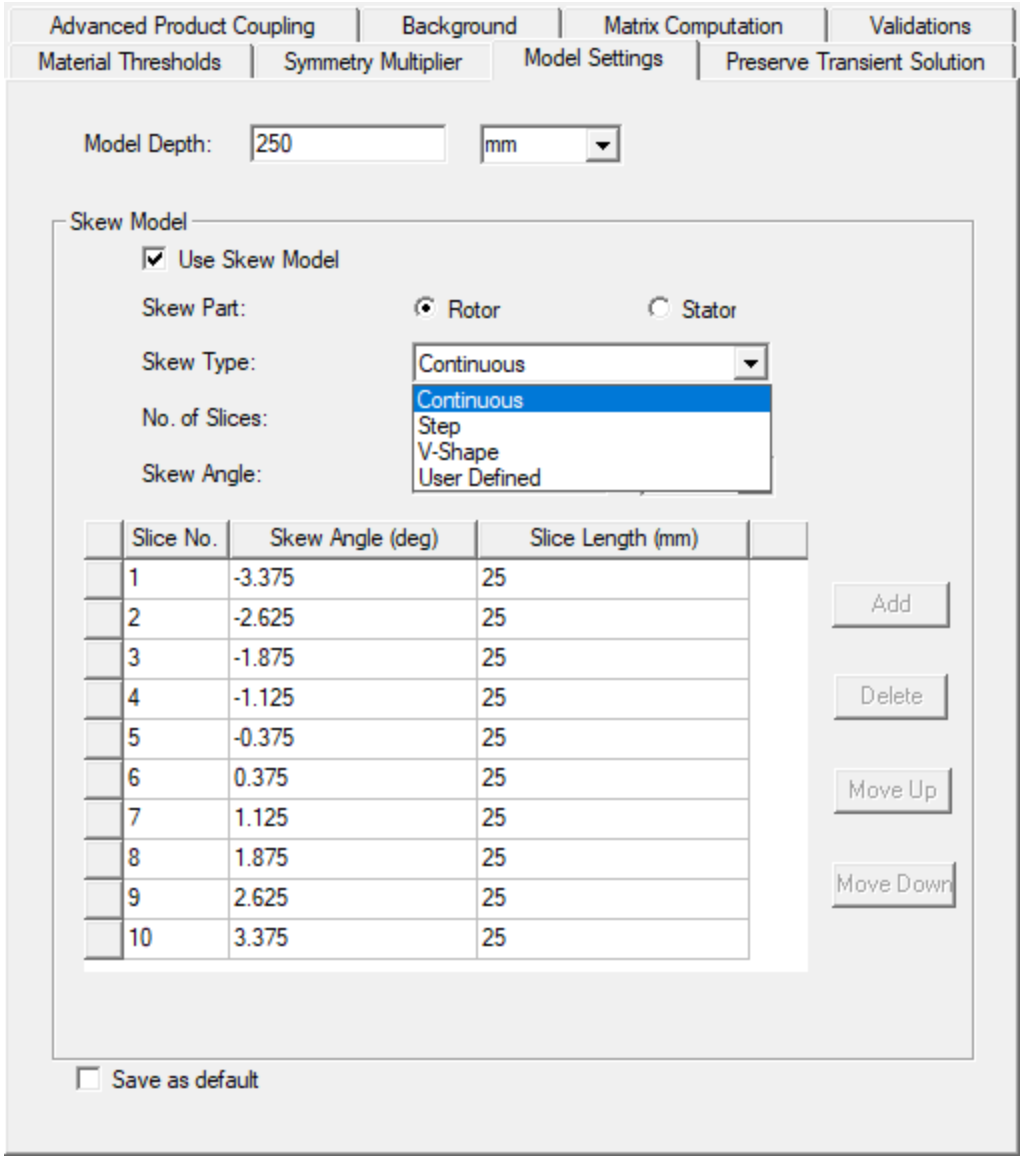
Maxwell 2D supports object-based harmonic force calculation using a multi-slice skew model along the axial direction of rotating machines. (Element-based harmonic force calculation is not applicable to 2D multi-slice skew model.)The multi-slice skew model can improve the accuracy of Maxwell 2D simulation when the slot/teeth on the stator/rotor are skewed. By using a skew model,

2D vibration modeling in Ansys Mechanical is no longer valid; thus, 3D simulation must be performed instead by applying object-based harmonic force calculation slice-by-slice.

Skew Model Setup

Note	For detailed information on Skew Model settings, refer to the *Model Settings Tab* topic.

Skew model setup in Maxwell 2D is defined on the **Model Settings** tab of the **2D Design Settings** dialog as described in the *Model Settings Tab* topic.

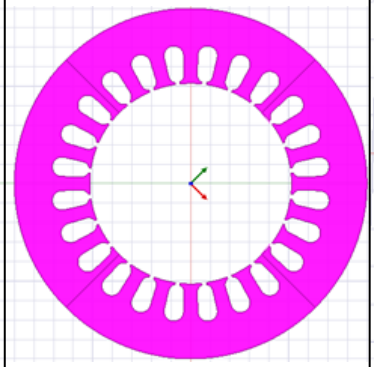


To create 3D geometry, the center position of each slice must be set properly:

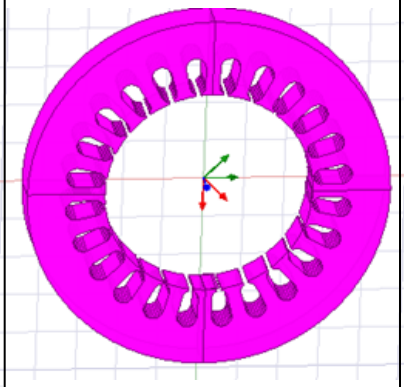
1. Create a 2D full model in Maxwell (do not unite half teeth at the M-S boundary), and generate 3D geometry of one slice; then create a Maxwell 3D design, and duplicate one slice in the Maxwell 3D.
2. Move the center of the 3D geometry to the origin of the global coordinate system; then set the Z axis as the rotating axis.
3. Rotate each slice to its individual circumferential position: slice angle is taken from the skew model settings (see figure above). The slice number indices are arranged from least to greatest, corresponding to Z coordinates of slice position (from least to greatest).

   For object-based harmonic force calculation, only one group of harmonic force components is generated for each selected object when skew model in not enabled in Maxwell 2D transient solver; and the Z coordinate of the object is set to zero by default. The “zero” Z position is used to match the selected geometry in Mechanical if coupled with the Harmonic response 3D solver.

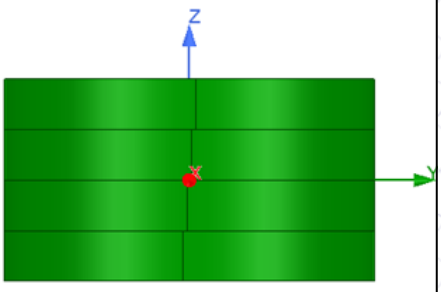
Maxwell 2D Partial Model



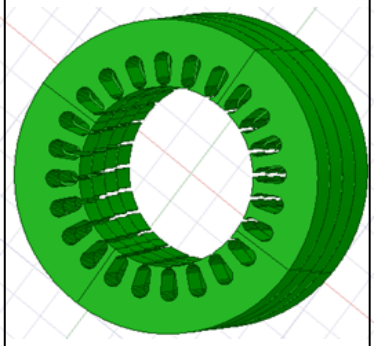
2D full model (stator only)



3D geometry of stator (one slice)



3D geometry after moving and rotating



3D geometry in harmonic response solver

Harmonic force calculation requirements for the Maxwell 2D slice model

Because the harmonic force calculated from the 2D slice model will be applied to a 3D model in Ansys Mechanical, it is very important to have a rule to link the skew angle to the slice number, and the slice number to the z-position in the Maxwell 2D solver. The relationship between slice number and z-position will determine how the 3D model will be generated.

The general rule is as follows:

- The sequence of slice number: from least to greatest with z position (this is for customer, not in the solver)
- The z-position: from least to greatest
- The skew angle: one-to-one with slice number, from slice 1 to slice N

The user generates 3D model based on skew angle, slice by slice, from least z-position to greatest z-position.

Related Topics

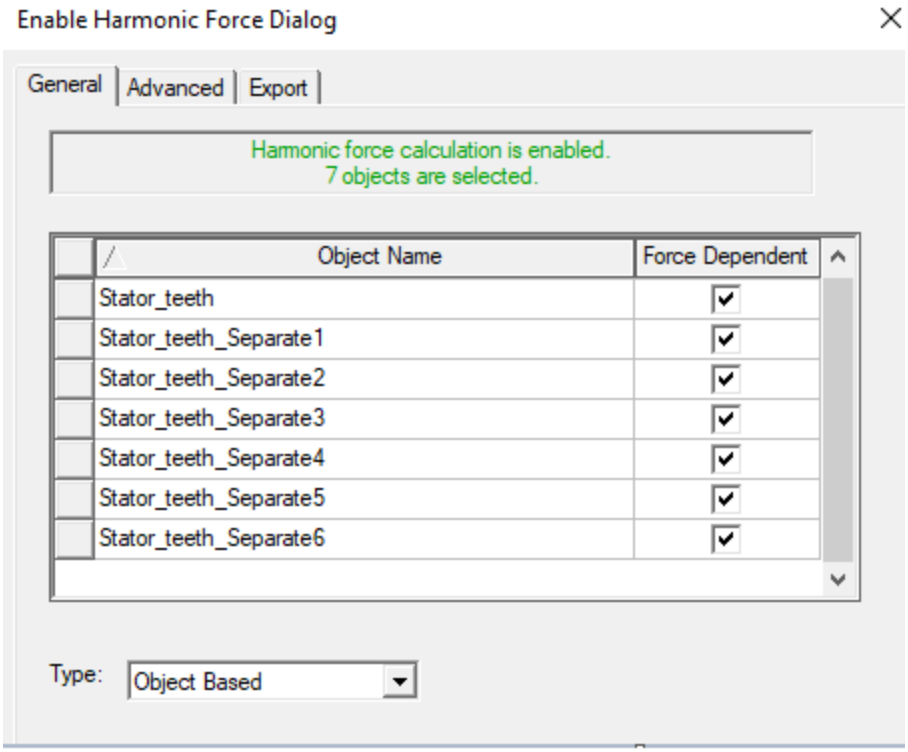
"Maxwell to Ansys Object-based Harmonic Force Coupling" on page 17-36

Maxwell to Ansys Element-based Harmonic Force Coupling

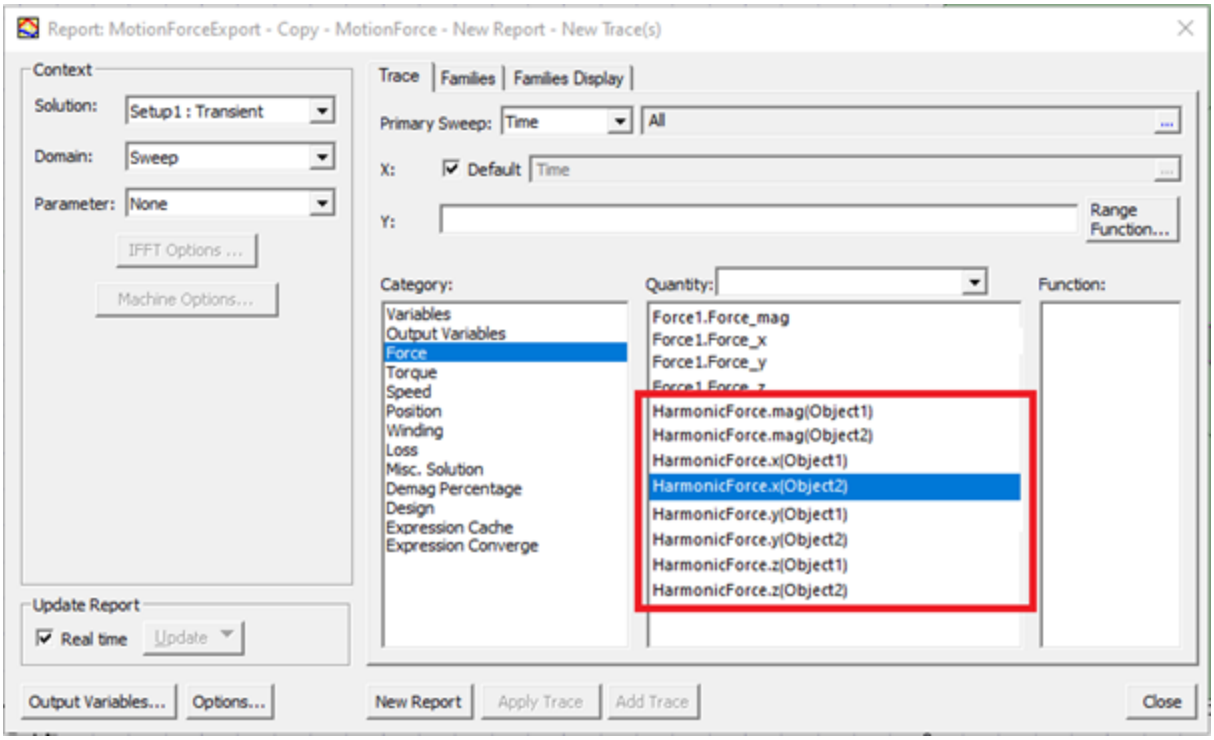
Transient Report of Object-based Harmonic Force Components

When object-based harmonic force is activated, the transient force data of selected harmonic force objects are available for plot in Maxwell transient solvers. The force component data is recorded at every time step, starting from time zero. To create a transient report of object-based harmonic force components:

1. In the **Enable Harmonic Force Calculation** dialog box, on the **General** tab, select **Object Based** as the harmonic force **Type**. Then check the **Force Dependent** objects for which force calculations are desired.

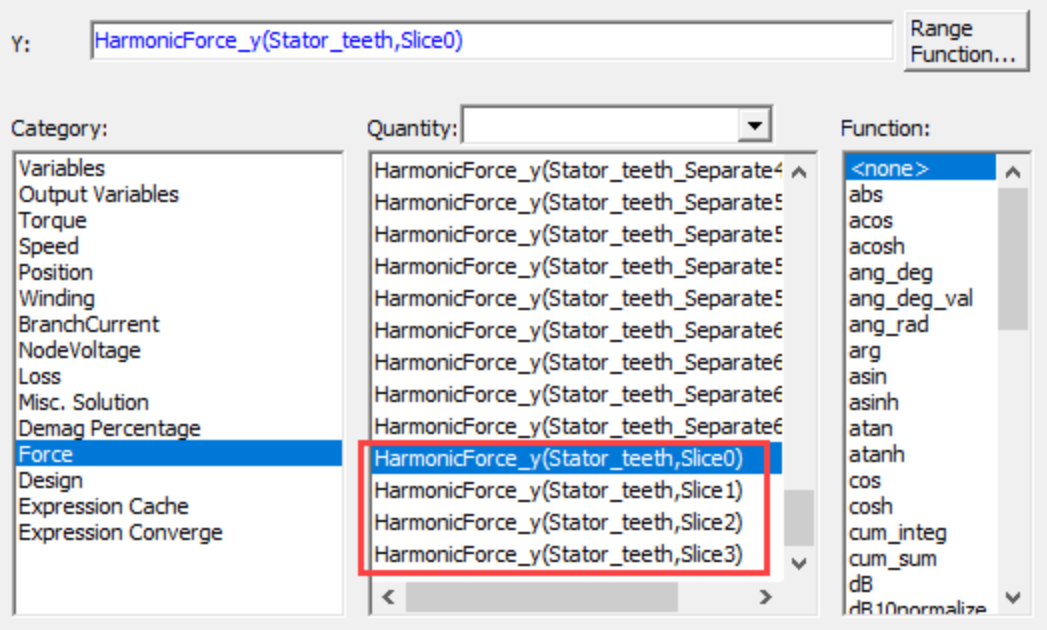


2. Create a **Transient Report** under **Results** in the project tree.



3. Select **Force** as the **Category**, and then select the harmonic force components you want to plot:
    - For 2D (XY) designs: **HarmonicForce_mag**, **HarmonicForce_x**, **HarmonicForce_y**.
    - For 3D designs: **HarmonicForce_mag**, **HarmonicForce_x**, **HarmonicForce_y**, **HarmonicForce_z**.
    - For 2D (RZ) designs: **HarmonicForce.z**.

Transient force reports also support Maxwell 2D with skew slice model. The transient force is calculated on each slice, and there are *n* forces available to be plotted for each selected object.



Note	For detailed information on Skew Model settings, refer to the *Model Settings Tab* topic.

Related Topics

"Maxwell to Ansys Object-based Harmonic Force Coupling" on page 17-36

Maxwell to Ansys Element-based Harmonic Force Coupling

Maxwell provides element-based solutions for one-way harmonic force coupling via Ansys Workbench schematic for:

- 2D/3D Eddy Current design element-based harmonic force coupling
- 2D/3D Transient design element-based harmonic force coupling

For detailed information on setting up and using harmonic force coupling in Workbench, refer to the Workbench help topics: *Importing Data into a Harmonic Analysis* and *Importing Data into a Thermal or Structural (Static or Transient) Analysis*

Related Topics

"Maxwell to Ansys Object-based Harmonic Force Coupling" on page 17-36
Element-based Harmonic Force Coupling for Eddy Current Designs
Element-based Harmonic Force Calculations and Coupling for Transient Designs

Element-based Harmonic Force Calculations and Coupling for Transient Designs

Maxwell 2D/3D supports *element-based* one-way harmonic force coupling for *transient* designs via the Workbench schematic. Maxwell enables the user to select multiple objects for force calculations for harmonic coupling with Workbench.

- For element-based (Surface) harmonic coupling, the force density is calculated at each surface element of the selected objects at every time step in the specified time window. Then, Discrete Fourier Transform is performed to transfer data from the time domain to the frequency domain. In the WB Harmonic response solver, surface force density is applied as surface pressure to the selected surfaces. There is no restriction on surface selection. Mapping is involved and it is done by post processor. Magnetostriction is not included for surface element-based harmonic force.
- For element-based (Volumetric) harmonic coupling, the force density is calculated at each element of the selected objects at every time step in the specified time window. Then, Discrete Fourier Transform is performed to transfer data from the time domain to the frequency domain. In the WB Harmonic response solver, volumetric force density is applied as body force to the selected objects. There is no restriction on object selection. Mapping is involved and it is done by post processor. Magnetostriction can be included for volumetric element-based harmonic force.

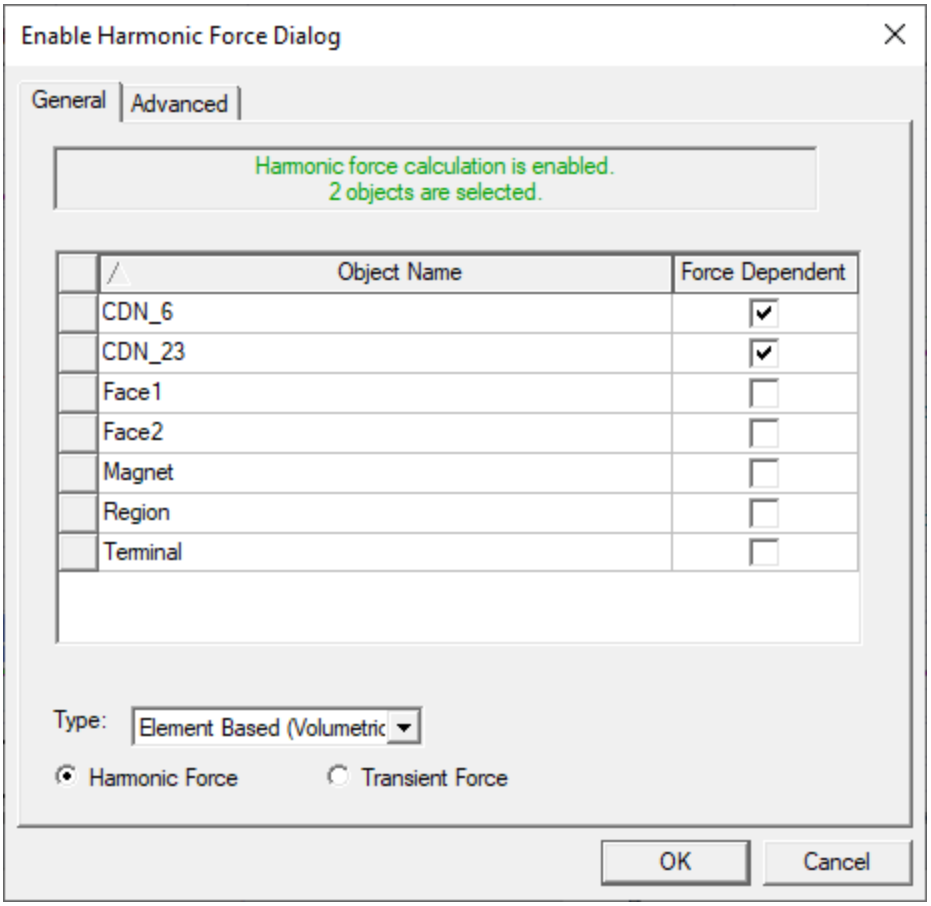
You can also setup multiple RPM Harmonic Force Coupling.

Note	Element-based harmonic force calculations can also be enabled for stand-alone transient designs.

For detailed information on setting up and using harmonic force coupling in Workbench, refer to the Workbench help topics: *Importing Data into a Harmonic Analysis* and *Importing Data into a Thermal or Structural (Static or Transient) Analysis*.

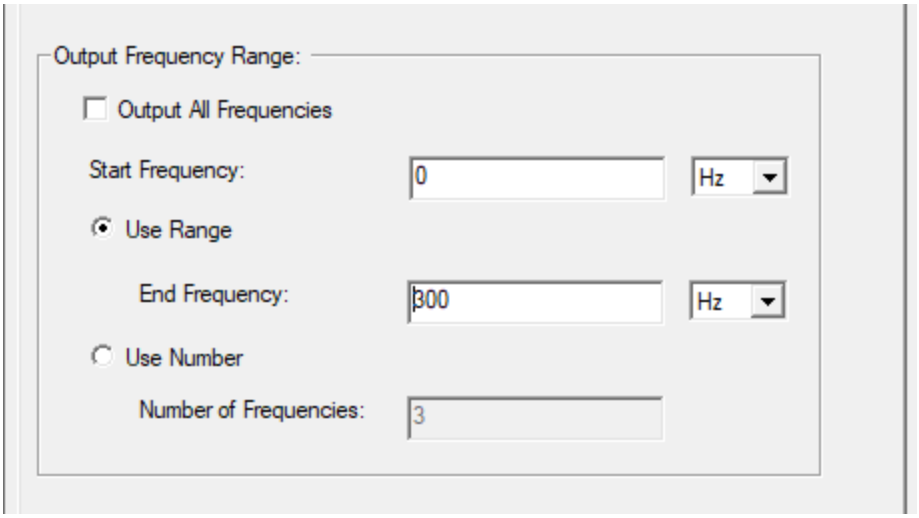
To enable element-based harmonic force calculations for transient designs in Maxwell:

1. Either select **Maxwell 2D** (or **Maxwell3D**)**>Enable Harmonic Force Calculation**, or right-click on the Design in the project tree and select **Enable Harmonic Force Calculation** to open the **Enable Harmonic Force Dialog** box.
    - For *transient* designs, the dialog has two tabs: **General**, and **Advanced**.
2. On the **General** tab, select **Element Based (Surface)** or **Element Based (Volumetric)** as the harmonic force **Type**.

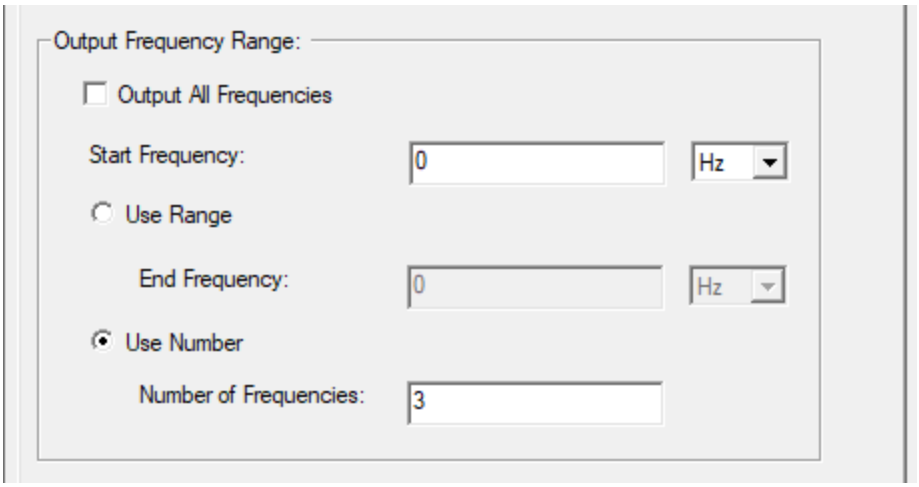


Check the **Force Dependent** objects for which force calculations are desired.

3. The **Advanced** tab provides additional settings to allow you to control collection of data from the force calculations.
    - For *2D and 3D transient* designs (either with rotational motion or with non-rotational motion), the **Output Frequency Range** panel allows you to specify the frequency range of calculated harmonic force components. The default is to **Output All Frequencies** that are calculated. There are two options to set the range of output frequencies:
        - **Use Range** – specify the **Start Frequency** and **End Frequency**. Parameterization of **Start Frequency** and **End Frequency** is supported.



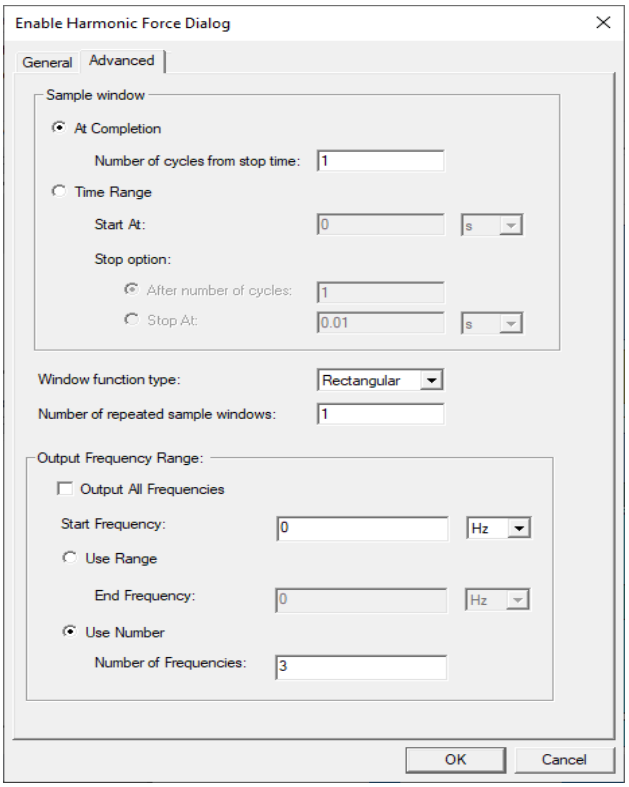
  - **Use Number** – specify the **Start Frequency** and the **Number of Frequencies**. Parameterization of **Number of Frequencies** is supported.



- For *transient* designs that include rotational motion:
  - The **Sample Window**'s collect samples **At completion** setting allows you to choose the **Number of cycles from stop time** for which data is collected. For example, if one cycle equals 1 second, and you solve for 3.5 seconds, setting the **Number of cycles from stop time** to 2 results in data being collected for the period from 1.5 seconds to 3.5 seconds. Alternatively, user can specify a **Time Range** to collect data, for which user can set a data collection **Start At** and a data collection **Stop option** – either stop data collection after a user-specified number of cycles (**After number of cycles**) or at a user-specified **Stop At** time.

- The **Window function type** options setting is used for Discrete Fourier Transform (DFT) of harmonic forces. Please refer to *Window Function Options for DFT of Harmonic Forces* for details on this setting.
- In the frequency domain of harmonic force, the base frequency is determined by the range of time window of data collection (f=1/ΔT). In order to get lower base frequencies, one must increase the time window of data collection, which means increasing the simulation time. It is waste of resources to do so when the solution reaches steady state/periodic in a transient simulation. Instead, you can set the **Number of repeated sample windows** to expand the range of time window of data collection without increasing simulation time by repeating the collected data *n* times before doing DFT. By applying this option, you can get a lower base frequency and higher frequency resolution, but the magnitudes of force components are close to zero at the introduced frequencies

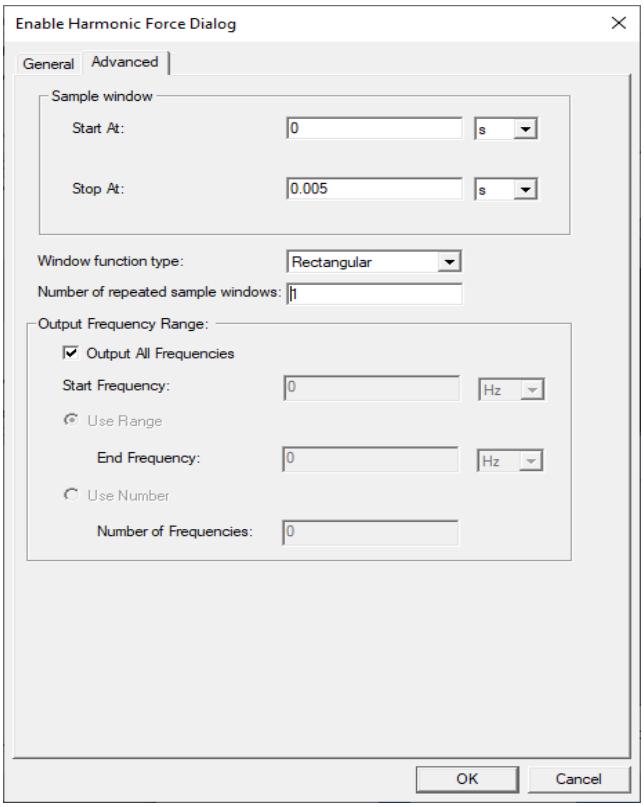
> **Note:** The **Number of repeated sample windows** setting is enabled only for the **Rectangular** (the default) **Window function type**.



- For *transient* non-rotational motion designs:
  - the **Sample Window** of the **Advanced** tab allows you to set the data collection **Start At** and **Stop At** times.

- The **Window function type** options setting is used for Discrete Fourier Transform (DFT) of harmonic forces. Please refer to *Window Function Options for DFT of Harmonic Forces* for details on this setting.
- You can set the **Number of repeated sample windows** to expand the range of time window of data collection.

> **Note:** The **Number of repeated sample windows** setting is enabled only for the **Rectangular** (the default) **Window function type**.



Once Maxwell is linked to the Workbench harmonic solver, it can export the harmonic force results to the Workbench Harmonic design. Harmonic transient force solutions are kept as part of Maxwell design solutions. If Workbench harmonic design settings are changed, the Maxwell solutions will be invalidated.

4. Click **OK** to close the dialog.

Related Topics

"Maxwell to Ansys Element-based Harmonic Force Coupling" on page 17-51

Force Densities in Maxwell

Force Density Calculation in Maxwell

Element-based Harmonic Force Coupling for Eddy Current Designs

For 2D/3D *eddy current* designs, Maxwell provides *element-based* solutions for one-way harmonic force coupling via Ansys Workbench. For element-based solutions, there is no need to select objects in Maxwell, because harmonic surface force is a default output in the solver.

For element-based (surface) coupling, there is a harmonic force vector on each surface node for objects. The user must select surfaces on the Workbench Mechanical side before these forces can be mapped. This mapping is done by the Maxwell post-processor.

For element-based (volumetric) harmonic coupling, the force density is calculated at each element of selected objects. The user must select bodies on the Workbench Mechanical side before these forces can be mapped. This mapping is done by the Maxwell post-processor.

For detailed information on setting up and using harmonic force coupling in Workbench, refer to the Workbench help topics: *Importing Data into a Harmonic Analysis* and *Importing Data into a Thermal or Structural (Static or Transient) Analysis*

Related Topics

"Maxwell to Ansys Object-based Harmonic Force Coupling" on page 17-36

Export Transient/Harmonic Force from Maxwell

There are four options to export transient forces and one option to export transient/harmonic force from Maxwell transient solvers:

- **LMS** – provides element-based transient surface node force and mesh to link to Virtual lab, only for stator of electric machine.
- **Romax** – provides object-based transient force for the stator of electric machines and torque information.
- **UNV** – UNV file format is a standard universal file format, and it can be read by most of third-party software. Element-based transient surface/volumetric node force will be outputted with mesh.
- **Ansys Motion** – provides object-based transient force, and element-based surface transient force to Ansys-Motion for structure and acoustic simulation.

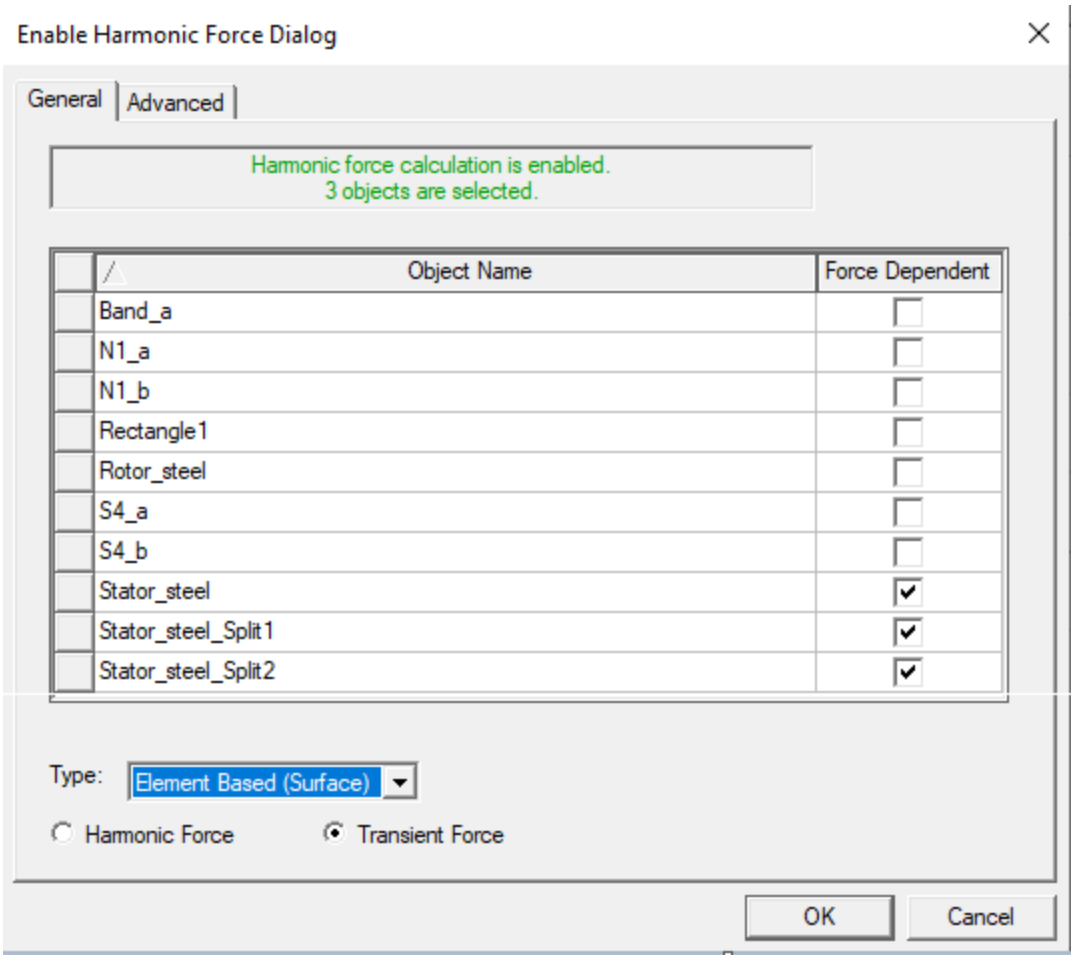
  > **Note:** Ansys Motion does not currently support harmonic force.

- **Ansys Mechanical**
  - For 3D transient solver, if **Element Based (Volumetric)** harmonic force selected, this option provides element-based volumetric harmonic force in `.csv` file format to *Ansys-Harmonic Response* for acoustic simulation.
  - If **Object Based** harmonic force is selected, this option provides object-based

harmonic force with RPM parametric sweep to generate waterfall diagram in *Ansys Mechanical*.

All forces and mesh are output with the full model. To enable “Export transient/harmonic force” from Maxwell, following steps must be done before solving the design:

1. Either select **Maxwell 2D** (or **Maxwell3D**)**>Enable Harmonic Force Calculation**, or right-click on the Design in the project tree and select **Enable Harmonic Force Calculation** to open the **Enable Harmonic Force Dialog** box.



2. On the **General** tab, select the **Force Dependent** objects for which force calculations are desired. For LMS, select only one object (stator).
3. Select **Type** of force for output. Choices are: **Object Based**, **Element Based (Surface)**, or **Element Based (Volumetric)**.
4. Check **Transient Force** if you want to export transient forces. (**Harmonic Force** is for Workbench Harmonic Response Coupling.)

> **Note:** Ansys Motion does not currently support harmonic force.

5. Check **Harmonic Force** if you want to export element-based volumetric harmonic force in `.csv` file format. This export option is only available for the Maxwell 3D magnetic transient solver.

> **Note:** Ansys Motion does not currently support harmonic force.

After solving the design, select **Maxwell 2D** (or **Maxwell3D**)**>Export Transient/Harmonic Force** to generate transient forces.

Related Topics

"Maxwell to Third-Party Coupling by UNV File " on page 17-63
"Maxwell to Romax Coupling " on page 17-61
"Maxwell to LMS Virtual.Lab Coupling " below
"Maxwell One-way Coupling with Ansys Motion " on page 17-65

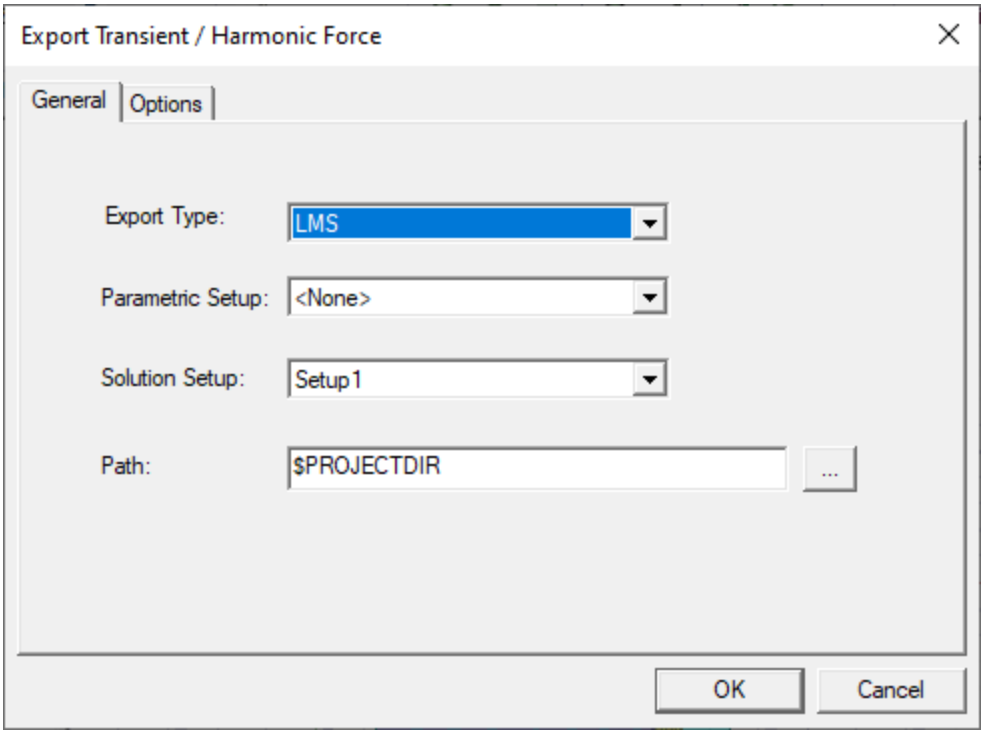
Maxwell to LMS Virtual.Lab Coupling

Maxwell supports one-way coupling with Siemens® LMS Virtual.Lab™.

To enable output of Maxwell solutions for use in Virtual.Lab:

1. Follow the steps in **Export Transient/Harmonic Force from Maxwell** to enable transient force output before solving the Maxwell design.
2. After solving, either select **Maxwell 2D** (or **Maxwell3D**)**>Export Transient/Harmonic Force**, or right-click on the Design in the project tree and select **Export Transient/Harmonic Force** to open the **Export Transient/Harmonic Force Dialog** panel.
3. On the **General** tab:
   - Select **LMS** for the **Export Type**.
   - Specify a **Parametric Setup** if any (If **<None>** is selected, only the nominal results will be exported).
   - Select a **Solution Setup**.

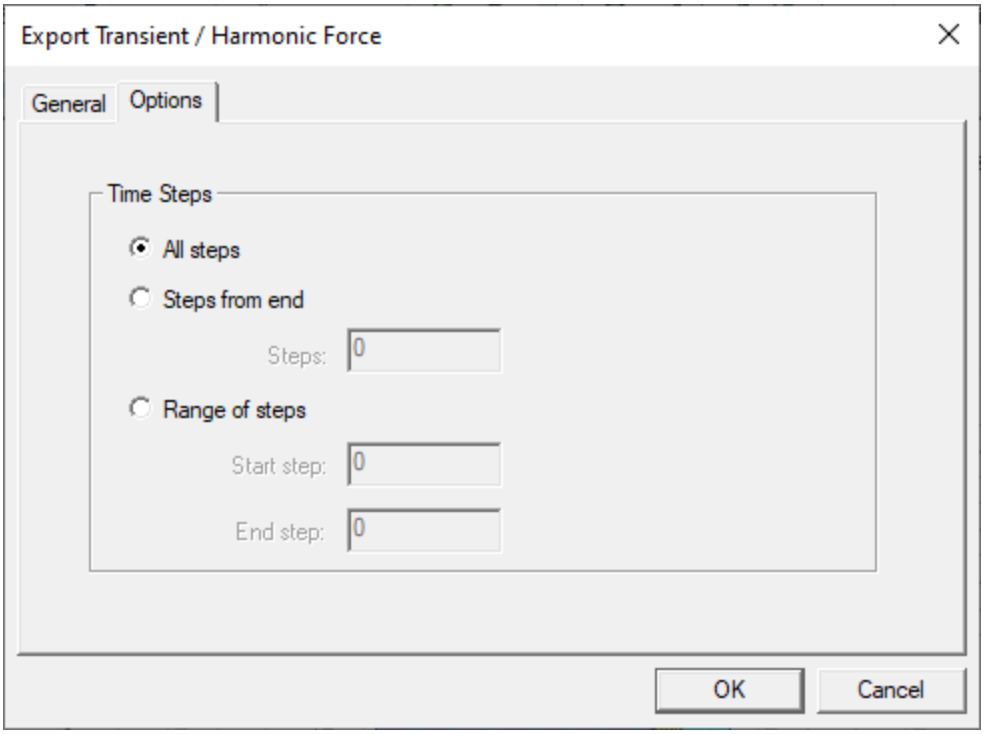
- Specify the path of the location of output file (the default is the project directory).



4. On the **Option** tab, settings allow you to control collection of data for export to Virtual.Lab. You can choose to include **All steps**. The **Steps from end** setting allows you to choose the number of **Steps** from the end for which data is collected.

   > **Note:** If rotation motion is defined in the project, the input number of steps will be ignored. Instead the last cycle will be the time window.

   Alternatively, you can select a **Range of steps**, for which you can set a data collection **Start step** and a data collection **End step**.



5. Click **OK** to close the dialog box. When you click **OK**, all the results (nominal or parametric solution) will be exported to the specified folder.

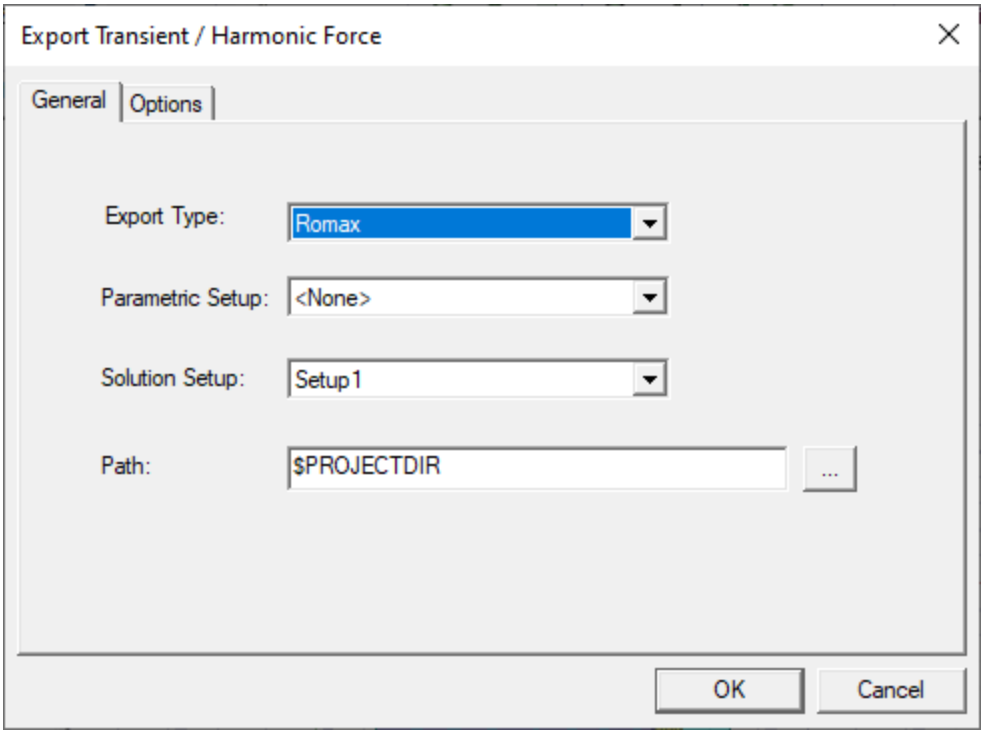
Maxwell to Romax Coupling

Maxwell provides object based transient force and torque components on selected objects. Maxwell supports one-way coupling with Romax.

To enable output of Maxwell solutions for use with Romax:

1. Follow the steps in **Export Transient/Harmonic Force from Maxwell** to enable transient force output before solving the Maxwell design.
2. After solving, either select **Maxwell 2D** (or **Maxwell3D**)**>Export Transient/Harmonic Force**, or right-click on the Design in the project tree and select **Export Transient/Harmonic Force** to open the **Export Transient/Harmonic Force Dialog** panel.
3. On the **General** tab:
   - Select **Romax** for the **Export Type**.
   - Specify a **Parametric Setup** if any (If **<None>** is selected, only the nominal results will be exported).
   - Select a **Solution Setup**.

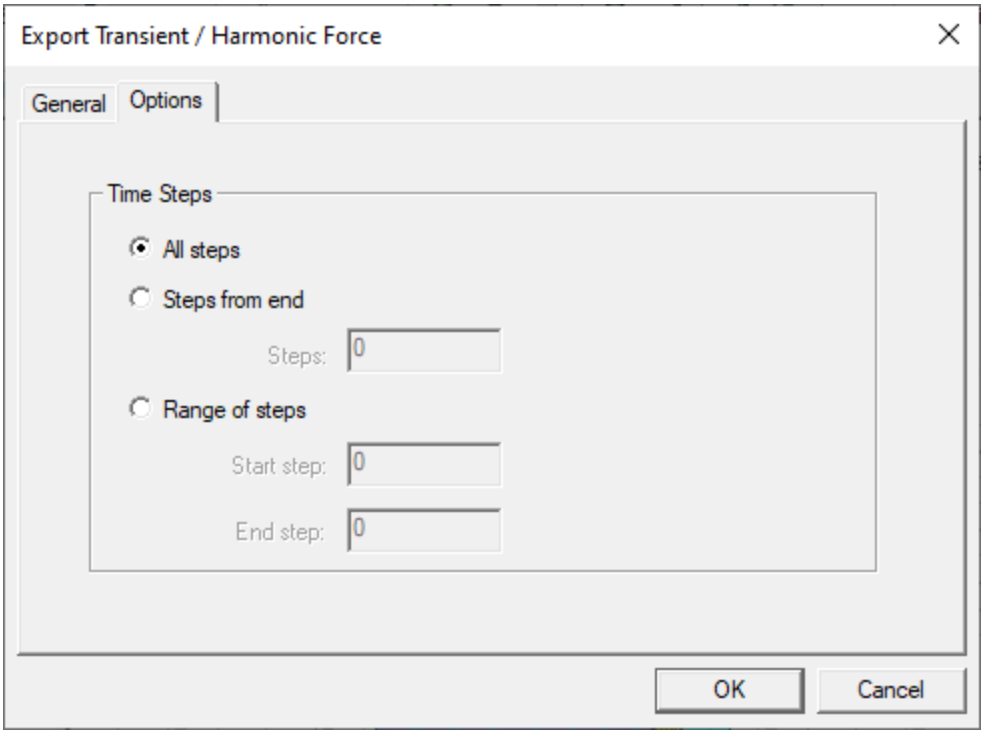
- Specify the path of the location of output file (the default is the project directory).



4. On the **Option** tab, settings allow you to control collection of data for export to Romax. You can choose to include **All steps**. The **Steps from end** setting allows you to choose the number of **Steps** from the end for which data is collected.

   > **Note:** If rotation motion is defined in the project, the input number of steps will be ignored. Instead the last cycle will be the time window.

   Alternatively, you can select a **Range of steps**, for which you can set a data collection **Start step** and a data collection **End step**.



5. Click **OK** to close the dialog box. When you click **OK**, all the results (nominal or parametric solution) will be exported to the specified folder.

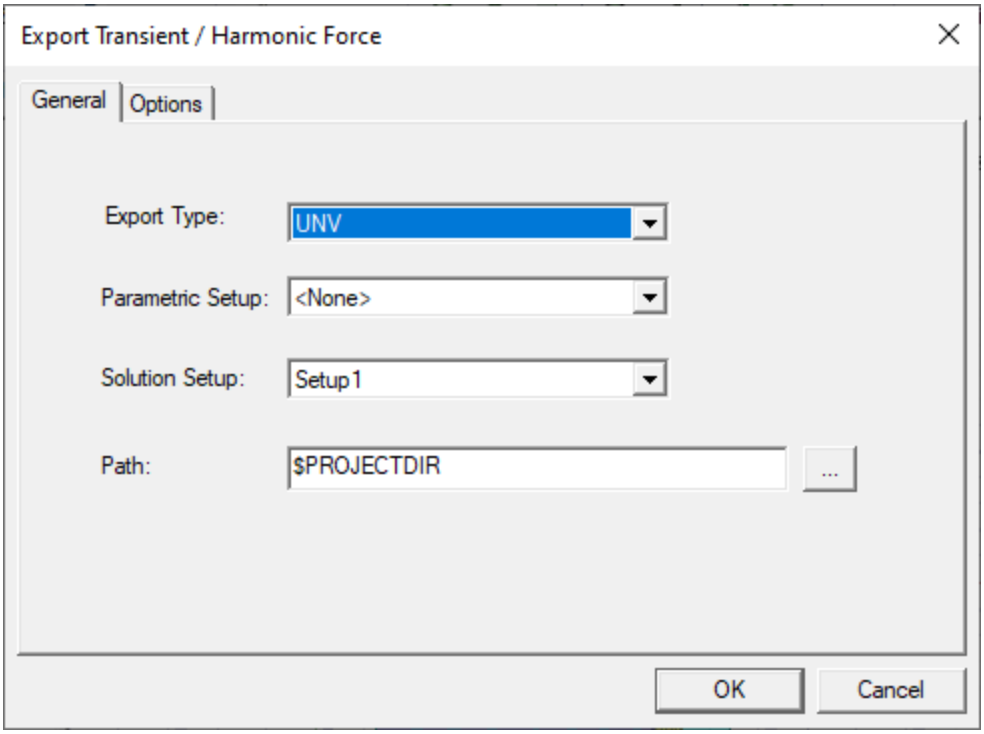
Maxwell to Third-Party Coupling by UNV File

Maxwell provides option of one-way transient force coupling with any third party through standard universal (**UNV**) file format.

To enable output of Maxwell solutions for use with any third-party through UNV file format:

1. Follow the steps in **Export Transient/Harmonic Force from Maxwell** to enable transient force output before solving the Maxwell design.
2. After solving, either select **Maxwell 2D** (or **Maxwell3D**)**>Export Transient/Harmonic Force**, or right-click on the Design in the project tree and select **Export Transient/Harmonic Force** to open the **Export Transient/Harmonic Force Dialog** panel.
3. On the **General** tab:
   - Select **UNV** for the **Export Type**.
   - Specify a **Parametric Setup** if any (If **<None>** is selected, only the nominal results will be exported).
   - Select a **Solution Setup**.

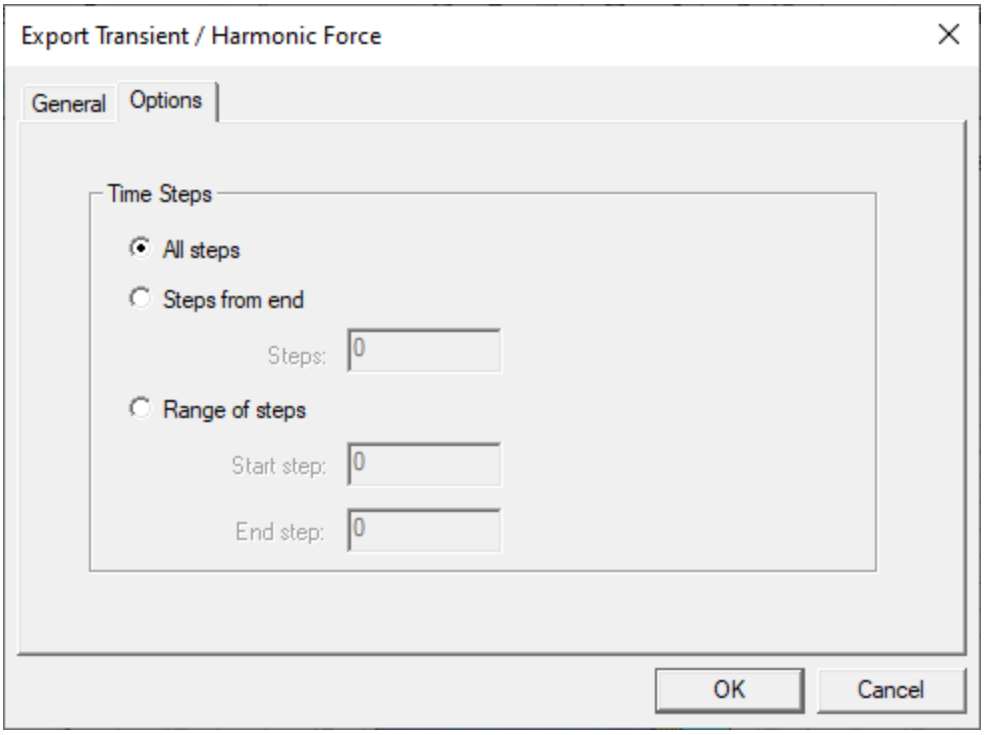
- Specify the path of the location of output file (the default is the project directory).



4. On the **Option** tab, settings allow you to control collection of data for export to a UNV file. You can choose to include **All steps**. The **Steps from end** setting allows you to choose the number of **Steps** from the end for which data is collected.

   > **Note:** If rotation motion is defined in the project, the input number of steps will be ignored. Instead the last cycle will be the time window.

   Alternatively, you can select a **Range of steps**, for which you can set a data collection **Start step** and a data collection **End step**.



5. Click **OK** to close the dialog box. When you click **OK**, all the results (nominal or parametric solution) will be exported to the specified folder.

Maxwell One-way Coupling with Ansys Motion

Maxwell exports transient force to couple with Ansys Motion for acoustic simulation. There are 2 options to link to Motion software in Maxwell:

- Object-based: provides object based transient force and torque components on selected objects.
- Element-based (surface): provides element-based transient surface node force and mesh.

All forces and mesh are output with full model.

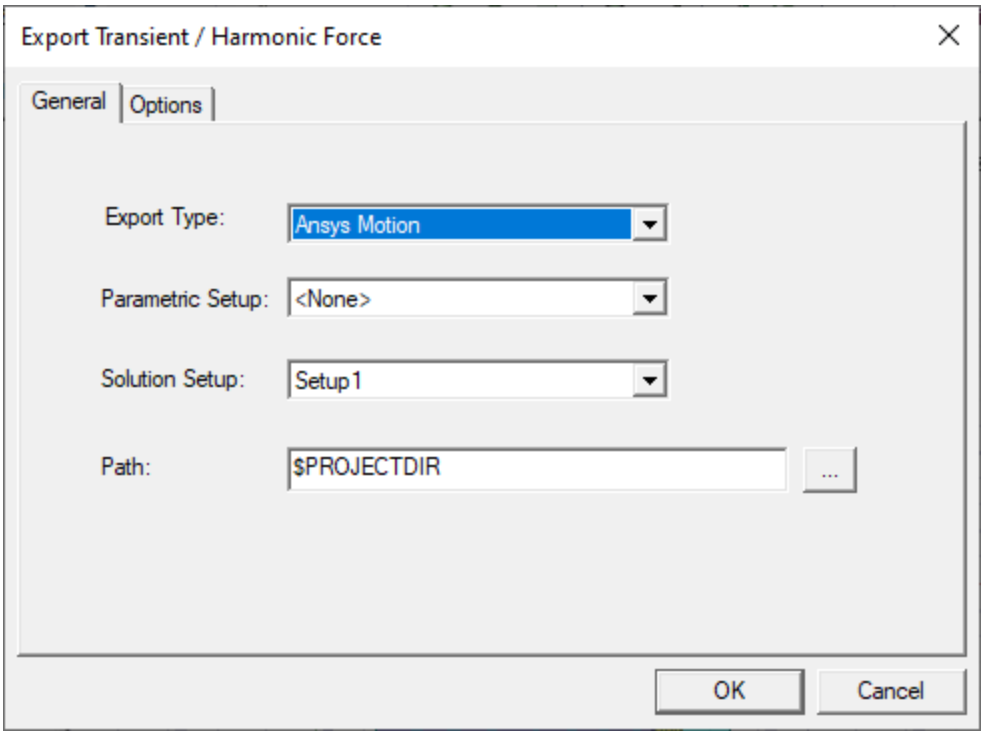
To enable output of Maxwell transient force solutions for link with Ansys Motion:

1. Follow the steps in **Export Transient/Harmonic Force from Maxwell** to enable transient force output before solving the Maxwell design.
2. After solving, either select **Maxwell 2D** (or **Maxwell3D**)**>Export Transient/Harmonic Force**, or right-click on the Design in the project tree and select **Export Transient/Harmonic Force** to open the **Export Transient/Harmonic Force Dialog** panel.

> **Note:** Ansys Motion does not currently support harmonic force.

3. On the **General** tab:

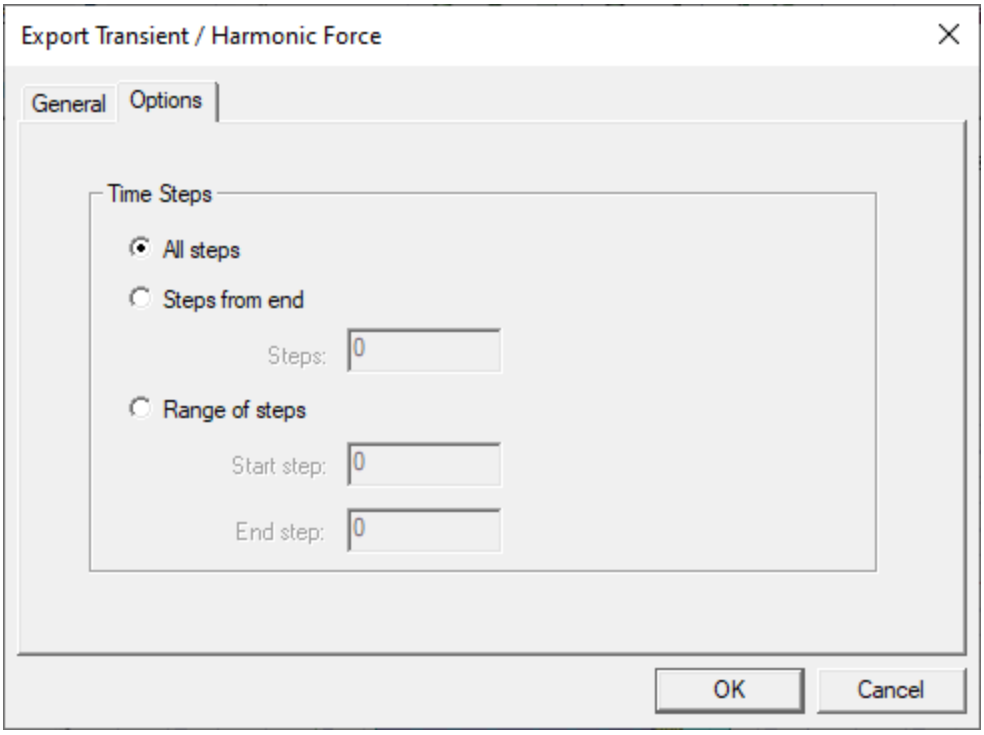
- Select **Ansys Motion** for the **Export Type**.
- Specify a **Parametric Setup** if any (If **<None>** is selected, only the nominal results will be exported).
- Select a **Solution Setup**.
- Specify the path of the location of output file (the default is the project directory).



4. On the **Option** tab, settings allow you to control collection of data for export to UNV file (and .AMESH and .ANF files if element-based selected). You can choose to include **All steps**. The **Steps from end** setting allows you to choose the number of **Steps** from the end for which data is collected.

   > **Note:** If rotation motion is defined in the project, the input number of steps will be ignored. Instead the last cycle will be the time window.

   Alternatively, you can select a **Range of steps**, for which you can set a data collection **Start step** and a data collection **End step**.



5. Click **OK** to close the dialog box. When you click **OK**, all the results (nominal or parametric solution) will be exported to the specified folder.

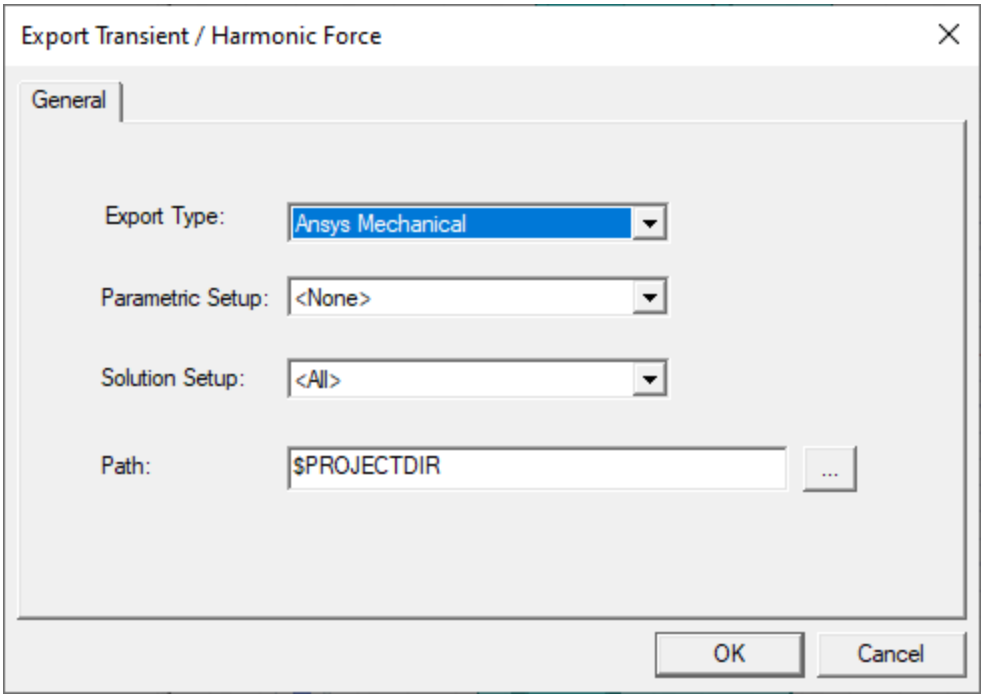
Maxwell One-way Coupling with Ansys Mechanical

Maxwell 2D and 3D transient designs can export object-based harmonic force to couple with Ansys Mechanical for acoustic simulation.

To enable output of Maxwell harmonic force solutions for link with Ansys Mechanical:

1. Follow the steps in **Maxwell to Ansys Object-based Harmonic Force Coupling** to enable object-based harmonic force output before solving the Maxwell design.
2. After solving, either select **Maxwell 2D** (or **Maxwell3D**)**>Export Transient/Harmonic Force**, or right-click on the Design in the project tree and select **Export Transient/Harmonic Force** to open the **Export Transient/Harmonic Force Dialog** panel.
3. On the **General** tab:
   - Select **Ansys Mechanical** for the **Export Type**.
   - Specify a **Parametric Setup** if any (If **<None>** is selected, only the nominal results will be exported).
   - All solved solution setups will be exported, so the **Solution Setup** pull-down list will always be **<All>**.

- Specify the path of the location of output file (the default is the project directory).

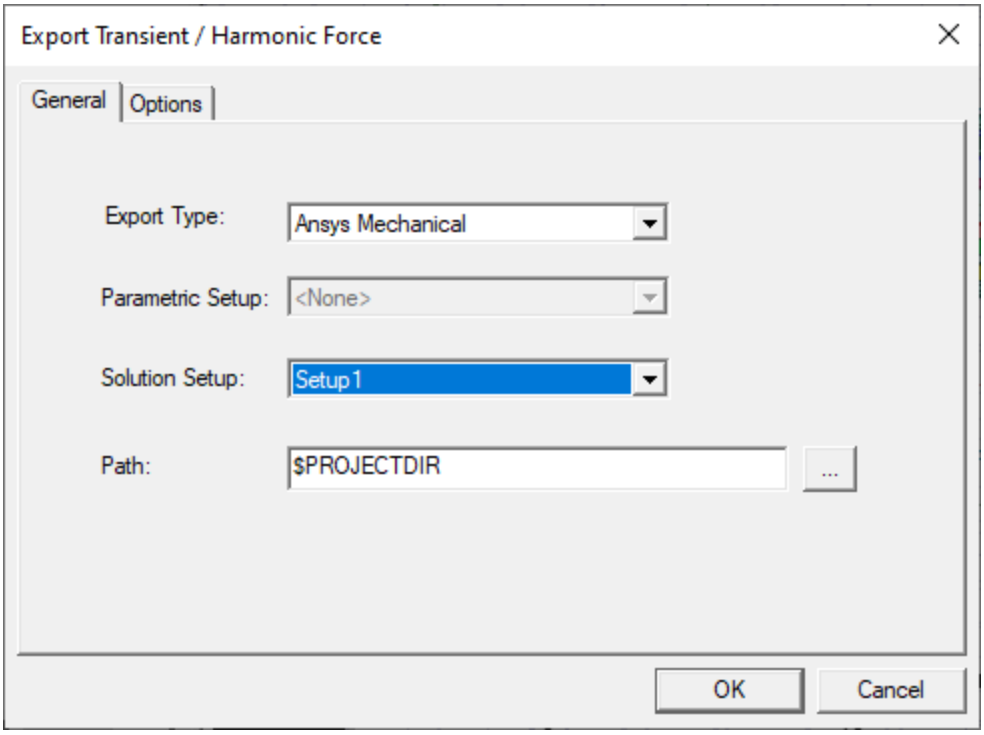


4. Click **OK** to close the dialog box. When you click **OK**, all the results (nominal or parametric solution) will be exported to the specified folder.

   You can then import the file into a WB harmonic component. For detailed information on setting up and using harmonic force coupling in Workbench, refer to the Workbench help topic: *Importing Data into a Harmonic Analysis*.

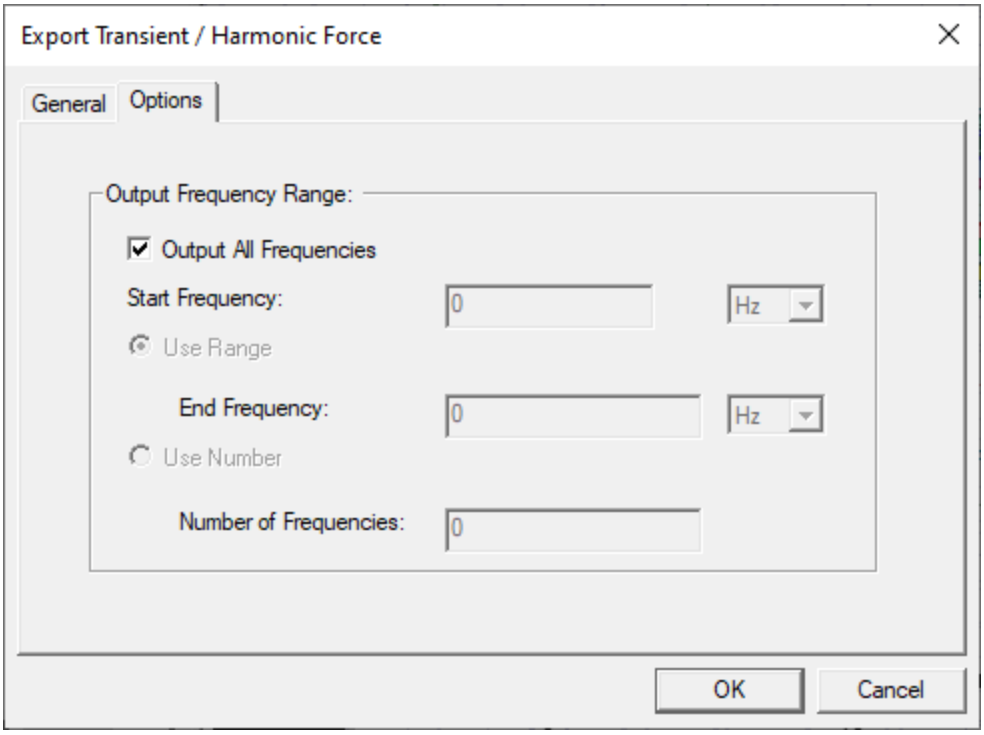
To enable output of Maxwell harmonic force solutions in a `.csv` file to couple with Ansys Mechanical:

1. Follow the steps in *Maxwell to Ansys Element-based Harmonic Force Coupling* to enable element-based harmonic force output before solving the Maxwell design.
2. After solving, either select **Maxwell3D>Export Transient/Harmonic Force**, or right-click on the Design in the project tree and select **Export Transient/Harmonic Force** to open the **Export Transient/Harmonic Force** dialog box.
3. On the General tab:
   - Select **Ansys Mechanical** for the **Export Type**.
   - Specify a **Parametric Setup** as **<None>**.
   - Select a **Solution Setup** in the pull-down list.
   - Specify the path of the location of output file (the default is the project directory).



On the **Options** tab, specify the **Output Frequency Range**. The file format in `.csv` file is (in line): Xc, Yc, Zc, Volume, Fx_re, Fx_im, Fy_re, Fy_im, Fz_re, Fz_im, Frequency1, Fx_re, Fxim, ……., Frequency2, ……

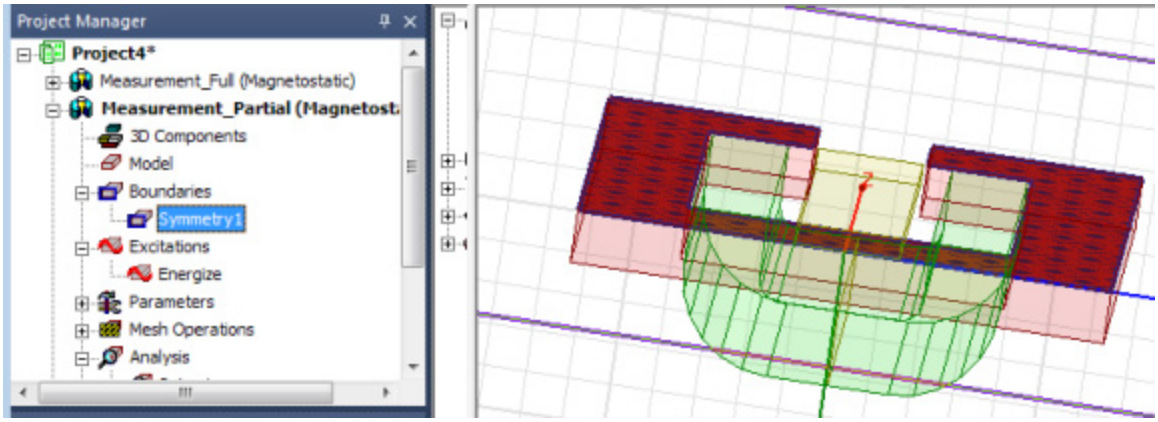
> **Note:**
>
> The frequency 0 (DC component) will be exported for either of the following conditions:
>
> - for **Output All Frequencies**: the first frequency will be 0.
> - for **Use Range**: if the **Start Frequency** is zero.

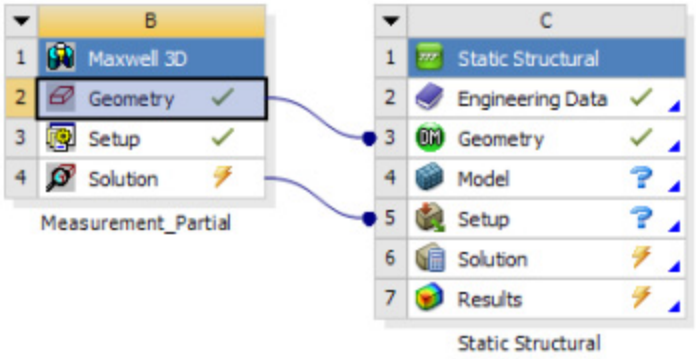


4. Click **OK** to close the dialog box. When you click OK, the `.csv` file will be exported to the specified folder. You can then import the file into a WB harmonic component.

Maxwell Partial Model Coupling with Workbench

This feature allows Maxwell 2D and 3D partial model coupling with Workbench applications such as Mechanical and Thermal. The full model geometry and solution can be extracted through symmetry from the partial model. Symmetry boundary conditions are used to solve problems with symmetry and, thus, allow the users to take advantage of the significantly reduced problem size for a given accuracy.

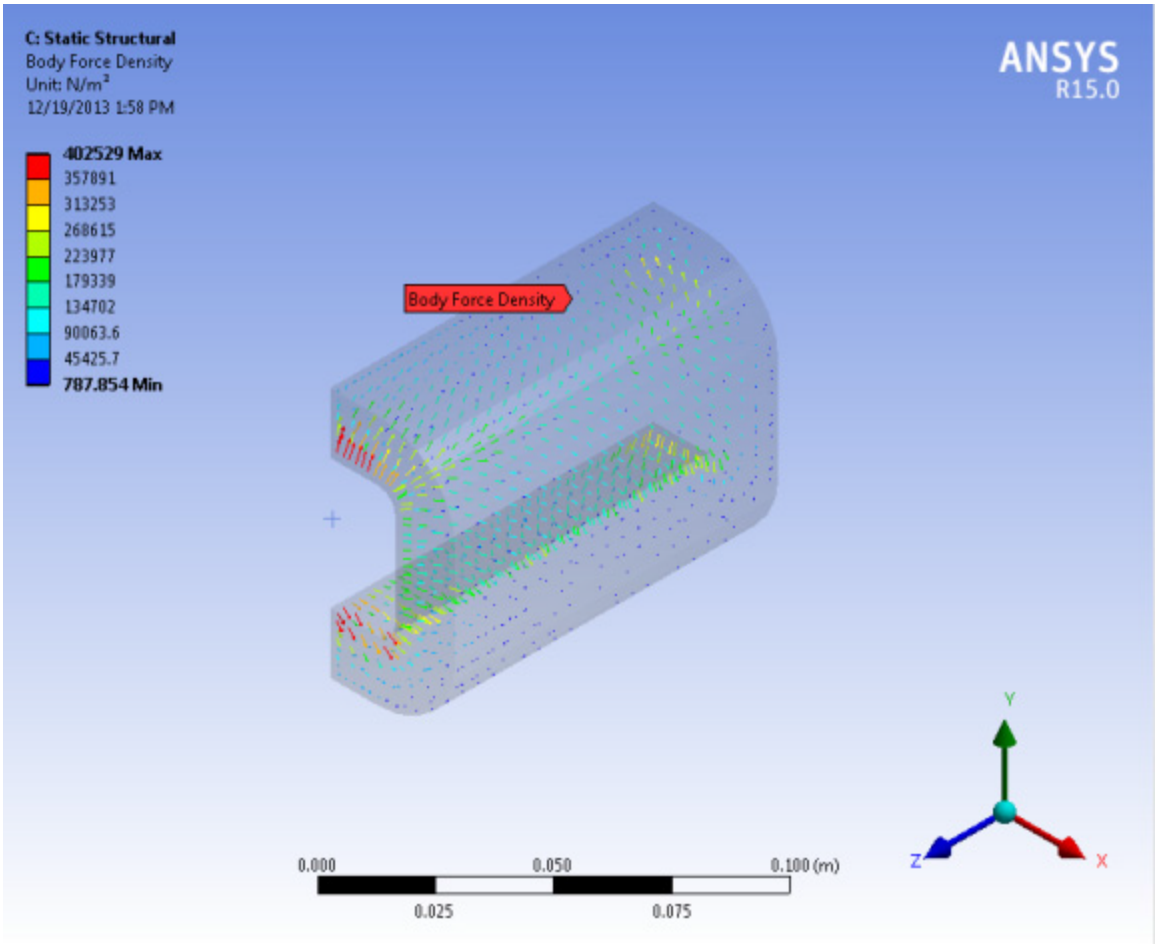




The following conditions must be observed for the Maxwell partial model:

- The model must be physically correct.
- Independent and dependent boundaries must be planar. If there is more than one pair of independent and dependent boundaries, all independents are coplanar and all dependents are coplanar.
- There are at most three mirroring symmetries, which are orthogonal to each other. If there is a pair of independent and dependent boundaries, then only one mirroring symmetry is allowed and it must be perpendicular to the axis of the independent and dependent boundaries.
- When the Maxwell side is a partial model and the Mechanical side is full model, full model loss / force will be mapped to the Mechanical side.
- When the Maxwell side is a partial model and the Mechanical side is also a partial model, only partial model loss / force will be mapped to the Mechanical side.

The user must ensure that the Maxwell partial model is physically correct, and that the full model in Workbench Mechanical, Thermal, etc. matches the partial model. The loss/force data from the partial Maxwell model is then mapped to the full model in Workbench Mechanical, etc. for solving. The mapping is handled implicitly.

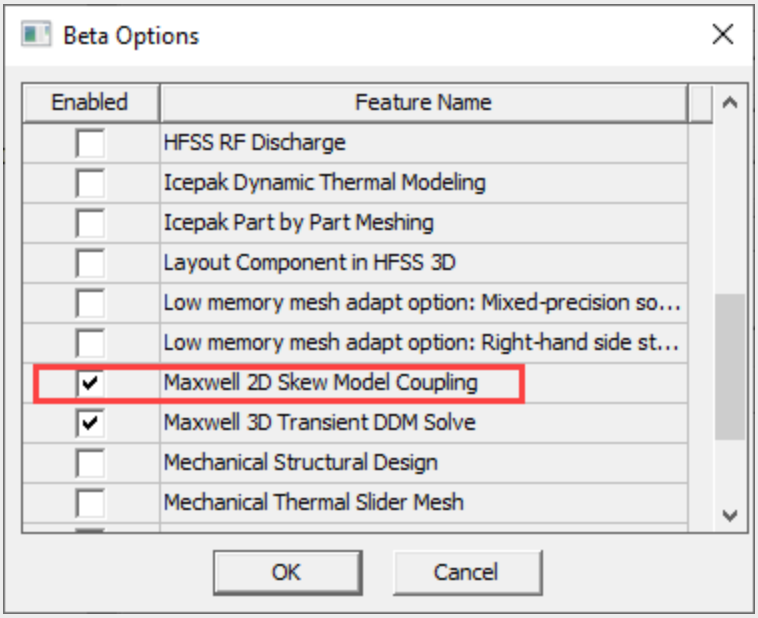


Related Topics

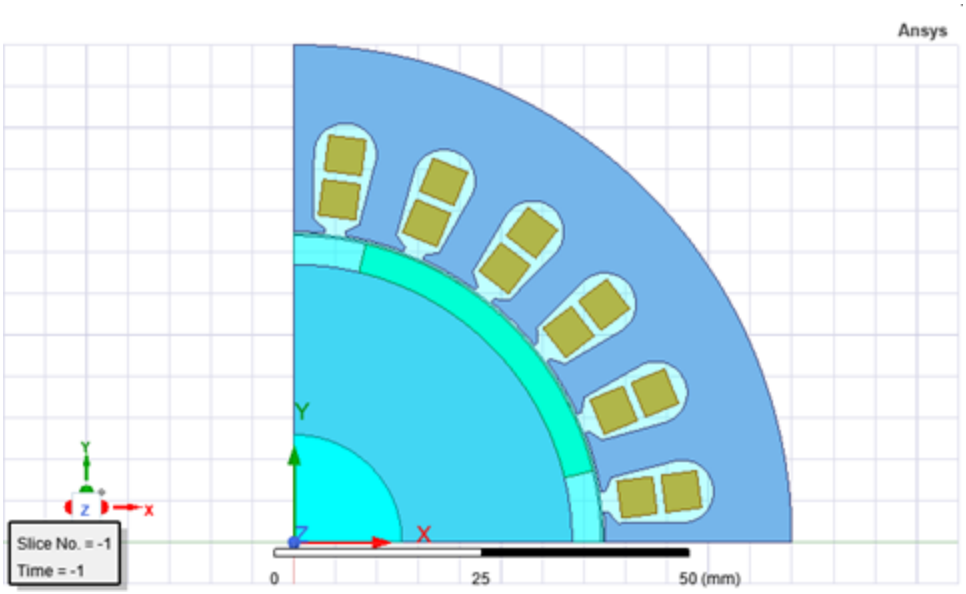
Symmetry for a Magnetostatic Field Solution
Matching (Independent/Dependent) Boundary for a Magnetostatic Field Solution
Symmetry for an Electric Field Solution
Matching (Independent/Dependent) Boundary for an Electric Field Solution
Symmetry for a Transient Solution
Matching (Independent/Dependent) Boundary for a Transient Solution
2D Symmetry Boundaries

[Beta] Element-based Multi-physics Coupling for 2D Multi-slice Skew Model

Note: You must enable this feature in General Options:Desktop Configuration before you can use it.



For Maxwell 2D transient skew models, this beta feature enables element-based multi-physics coupling to steady-state thermal and static/transient structural designs in Work bench. The target design could be 2D with matching geometry or 3D with equivalent geometry, i.e., extrusion of 2D geometry with matching skew type.



Currently, RMxprt can only generate corresponding 3D models from electric machines with the continuous skew type, and corresponding 3D designs cannot be automatically generated from 2D skew model designs.

The Maxwell 2D skew model itself can be a partial model with an independent/dependent boundary, in which case, the target could be either a partial or a full model. If the target geometry is 2D, data from the default slice (middle) will be mapped. If the target geometry is 3D, it should be an extrusion of the 2D geometry with matching skew type. Any geometry in 3D that is beyond the 2D model depth will not be mapped.

Limitations

- 3D geometry creation from 2D Skew models is not supported. This needs to be done manually and then couple the two designs.
- Stress feedback on Structural is not supported. Regarding 3D to 3D coupling, Maxwell-Structural coupling provides forces from Maxwell and receives displacement and stress from Structural.
- 2D to 3D EM coupling is not supported.

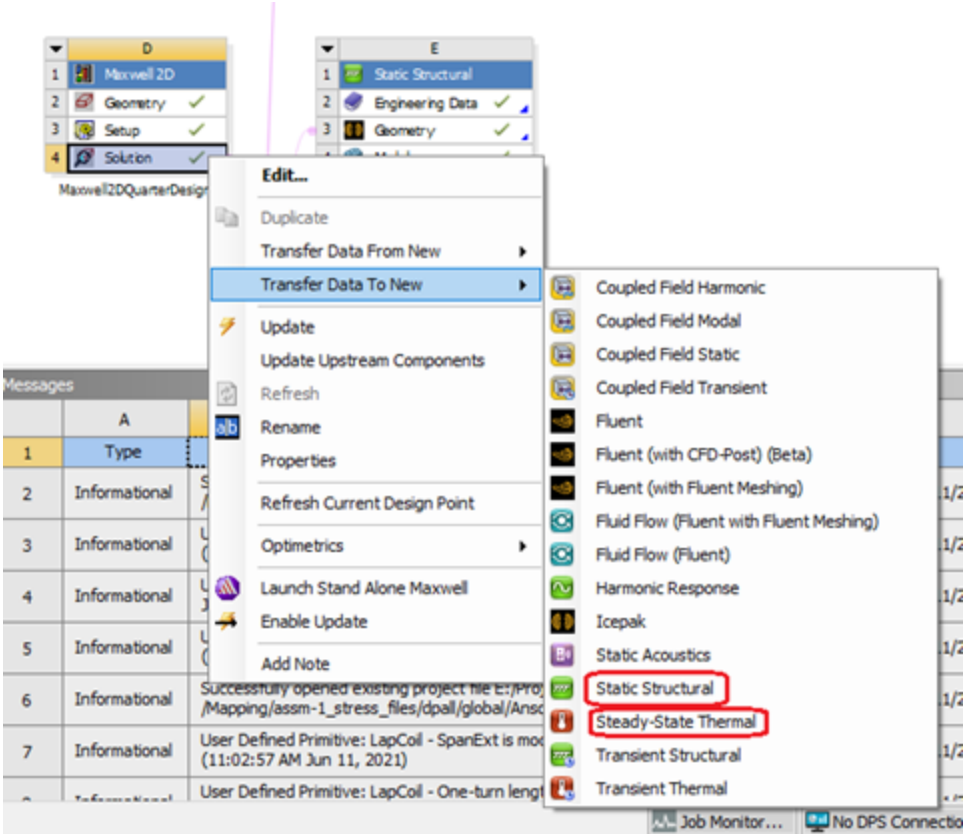
Use the following work flow:

1. From RMxprt, create a 2D skew model design and corresponding 3D design.

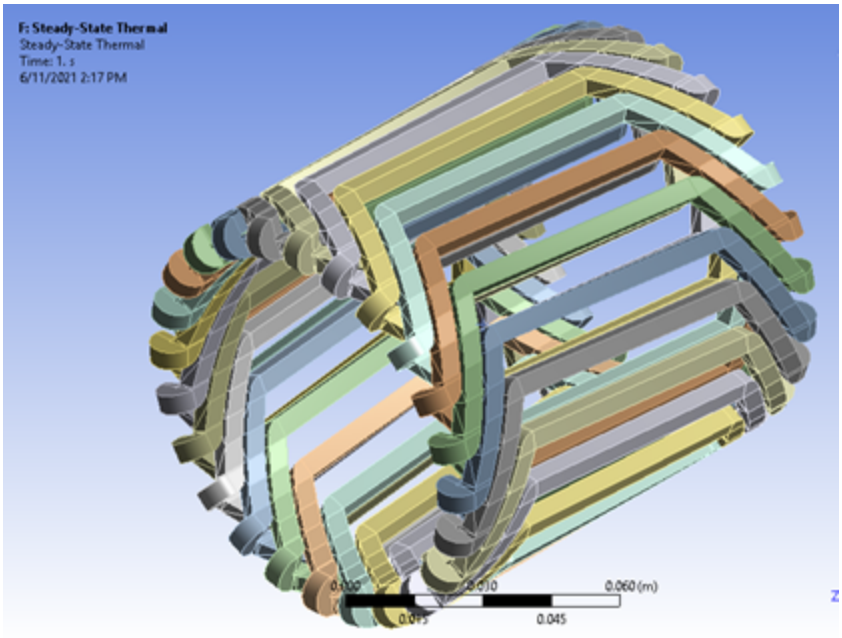
   > **Note:** For detailed information on Skew Model settings, refer to the *Model Settings Tab* topic.

2. Solve the 2D design and save the project.
3. Drag the project into Workbench.

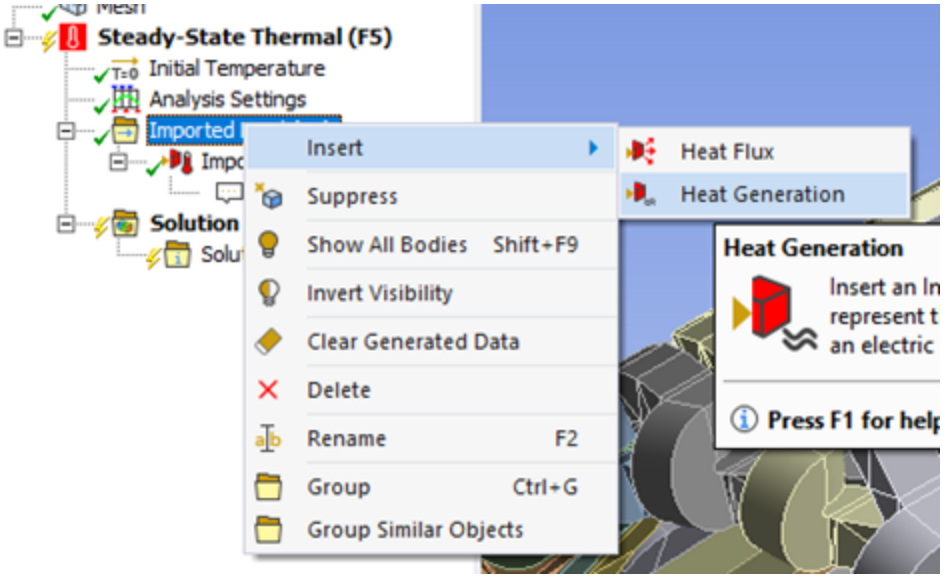
4. In the Maxwell 2D design, select **Solution>Transfer Data To New>Steady-State Thermal** or **Static Structural**.



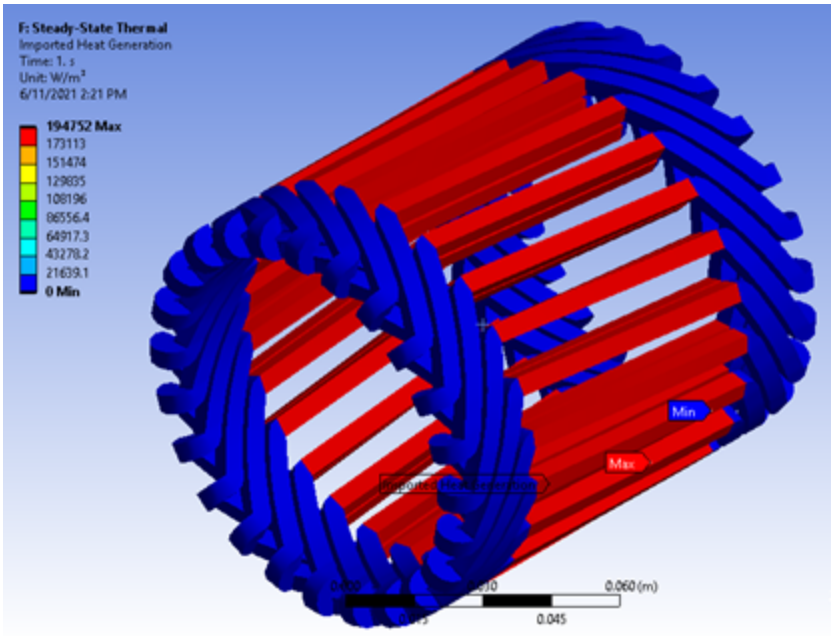
5. Drag the Geometry from the Maxwell 3D design to the Geometry of the newly created Mechanical design.
6. Refresh and/or Update the Geometry and Model in the Mechanical design.



7. In the Maxwell 2D, select **Solution>Update**.
8. In the Mechanical design, **Setup>Refresh**.
9. Double click **Setup**.
10. **Imported Load>Insert>Heat Generation**, select lossy objects and set end time.



11. Import the loads. Forces could be imported similarly.



Near Field Coupling between Maxwell and HFSS

The purpose of this coupling is to allow **Maxwell3D** frequency domain solution magnetic fields to be used as a near field source in the coupled **HFSS** design at exactly matched frequency.

Note	There is no **Far Field Wave** coupling datalink between **HFSS** and **Maxwell**.

The suggested sequence of steps to create the link between **Maxwell** and **HFSS** is as follows.

1. Create and solve the **Maxwell3D** frequency domain application at the chosen frequency.
2. Create the **HFSS** model.
3. Add "mapping" object to **HFSS** model.
4. Select all faces of the mapping object and apply a radiation boundary:
   - check the **Advanced Options** check box.
   - select the **Enforced field** radio button.
5. Click **HFSS>Excitations>Assign>Incident Wave>Near Field Wave**.

   The **Incident Wave** wizard appears. In the wizard:
   - Enter a **name** for the source or accept the default name and click **Next**;
   - In the **Incident Wave Source: Near Field Wave Options** dialog:
     a. Leave the translation and rotation data fields at their default zeros if the objects in the coupled designs have the desired location / orientation or change the values as needed;.

Note	You can change the location of the source design origin and orientation (rotation) to perform different **HFSS** simulations without re-solving the **Maxwell** design.

b. Click the **Setup Link** button to specify the location of the source, Maxwell design.

Select **OK** to exit the **Setup Link** dialog.

- Select **Finish** to finalize the Near Field link setup.

Magnetic Bias Source Coupling between Maxwell and HFSS

HFSS designs featuring ferrite materials can take advantage of non-uniform, static magnetic biasing fields to provide accurate design and simulation results. To achieve this goal it is possible to set up a coupling datalink between **HFSS** (target design) and a **Maxwell3D** magnetostatic design (source).

The suggested sequence of steps to create the link between the **HFSS** and **Maxwell3D** designs is as follows:

1. Create and solve the **Maxwell3D** design which contains the ferrite object and the corresponding non-uniform biasing fields.
2. Create the **HFSS** design which can have entirely different geometry but which contains the ferrite object(s) present in the linked **Maxwell3D** design;

Note	Make sure that in the coupled designs (**HFSS** and **Maxwell**) the ferrite objects have the same position and orientation in the global coordinate system.

3. In **HFSS** design, assign appropriate ferrite material property to object.
4. In **HFSS** design set the magnetic bias source (only one allowed per **HFSS** design):
   - Select the object(s) to be assign the non-uniform magnetic bias.
   - Click **HFSS>Excitation>Assign>Magnetic Bias** in the menu.

   The **Magnetic Bias Source** dialog appears:

   a. Enter a **Name** for the source.
   b. Select the **Non-Uniform** radio button.
   c. Click the **Setup Link** button to specify the applicable **Maxwell3D** project, design and solution for the link.

   Select **OK** when done.

   d. Select **Finish** to finalize the Magnetic Bias link setup.

Hint	It is possible to have the datalink create the **Maxwell** project automatically if it doesn't exist. When selecting the "Setup Link" button above, entering non-existent **Maxwell** project and design names will result in the creation of the needed **Maxwell** entities with the entire **HFSS** geometry included. All unneeded **HFSS** objects can be removed and the rest of the **Maxwell** model can be built around the ferrite object(s).

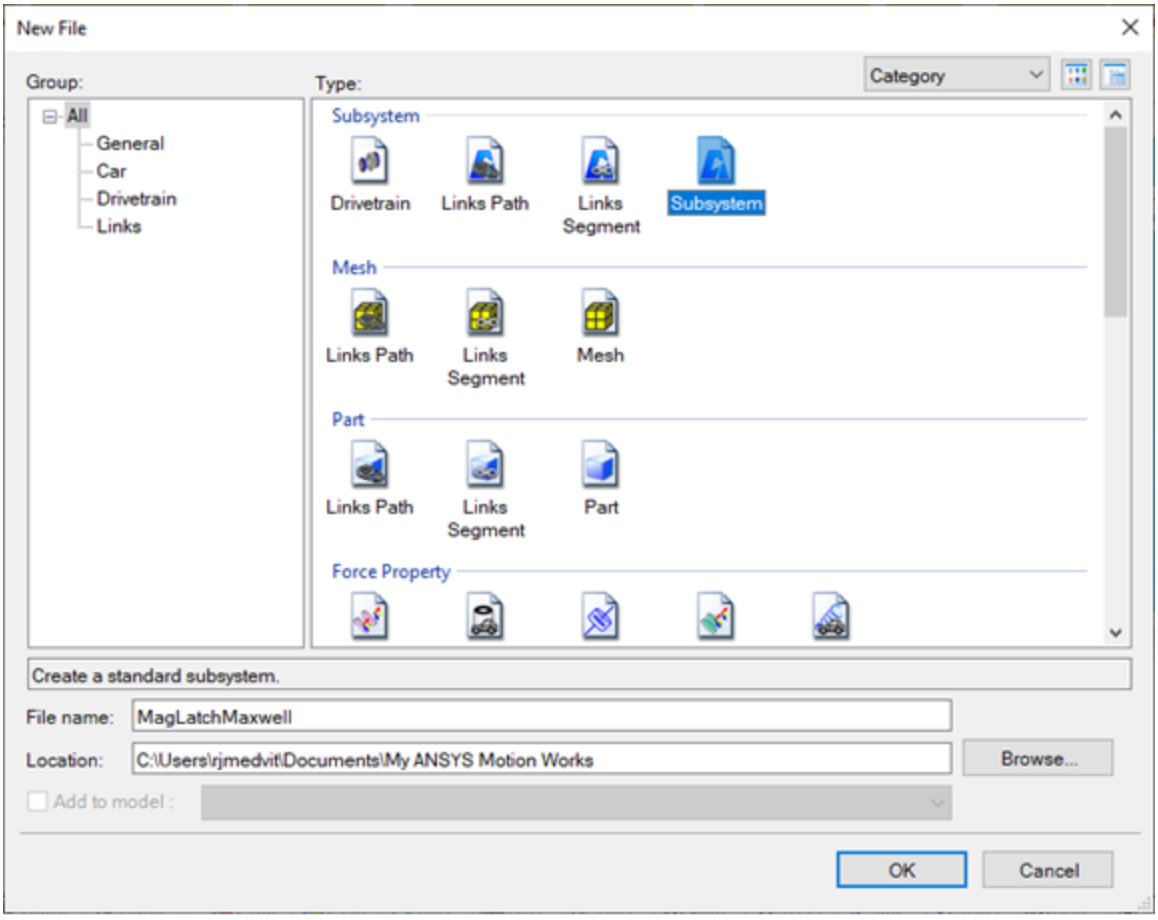
Magnetic Latch Model Co-simulation between Maxwell and Ansys Motion

Magnetic latch modeling is used typically for modeling magnetic latching structures such as magnetic chargers used by mobile devices. Magnetic latch modeling uses direct co-simulation between Maxwell and Ansys Motion. Ansys Motion is the primary (controlling) design, while Maxwell serves as the secondary design. During the co-simulation, Motion provides objects position data to Maxwell, then Maxwell feeds force/torque data for these objects back to Motion.

The suggested workflow for magnetic latch modeling using co-simulation between Ansys Motion and Maxwell is as follows:

> **Note:** Please refer to the *Ansys Motion Preprocessor Manual* installed with the Ansys Motion software for details on using the Motion Preprocessor for the following steps.

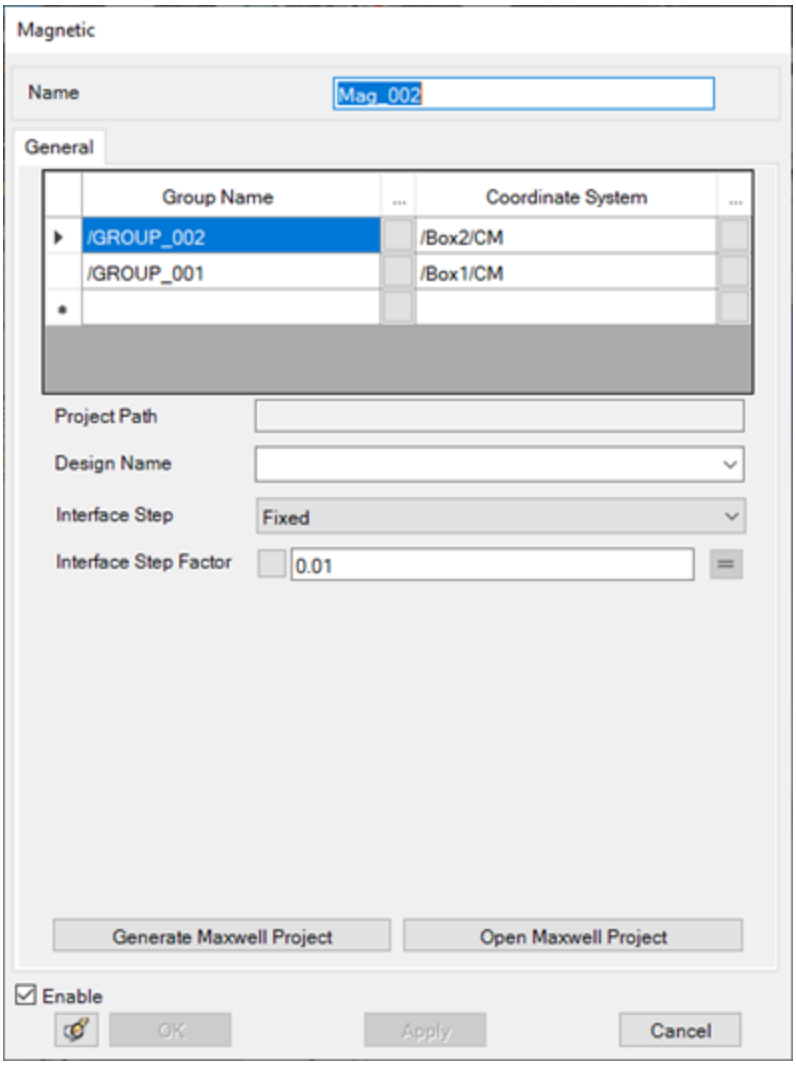
1. In **Ansys Motion Preprocessor**, create a new **Subsystem** file for which you specify a project path and project name.



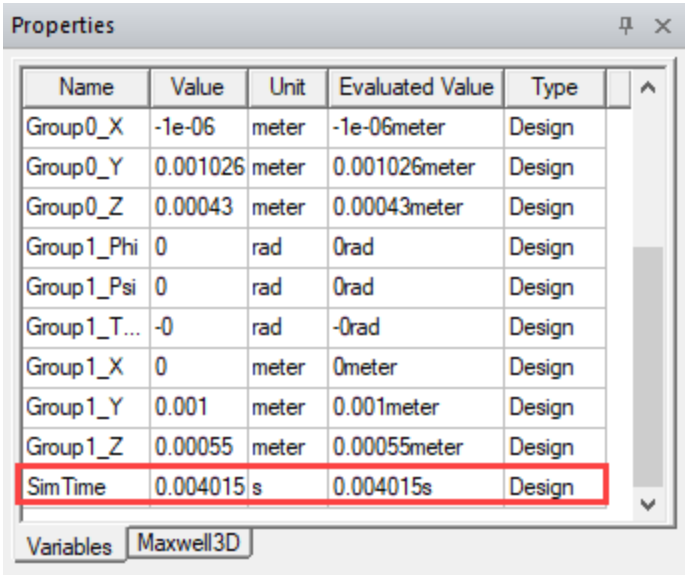
2. Import a CAD file of the model. You can modify object properties such as materials as needed.
3. After creating the model, **Group** objects of interest that have position changes and/or need force/torque calculations in the Maxwell design.

> **Note:** For performance considerations, it is recommended that you not perform force/torque calculations on all objects.

4. On the Ansys Motion Preprocessor **Force** tab, click the **Magnetic** button, pick the desired **Groups**, then click **OK**. In the **Magnetic** dialog box enter the desired **Interface Step Factor** (in seconds) or accept the default 0.01. This sets the Maxwell simulation rate. The **Coordinate System** assigned to each group defaults to center-of-mass.

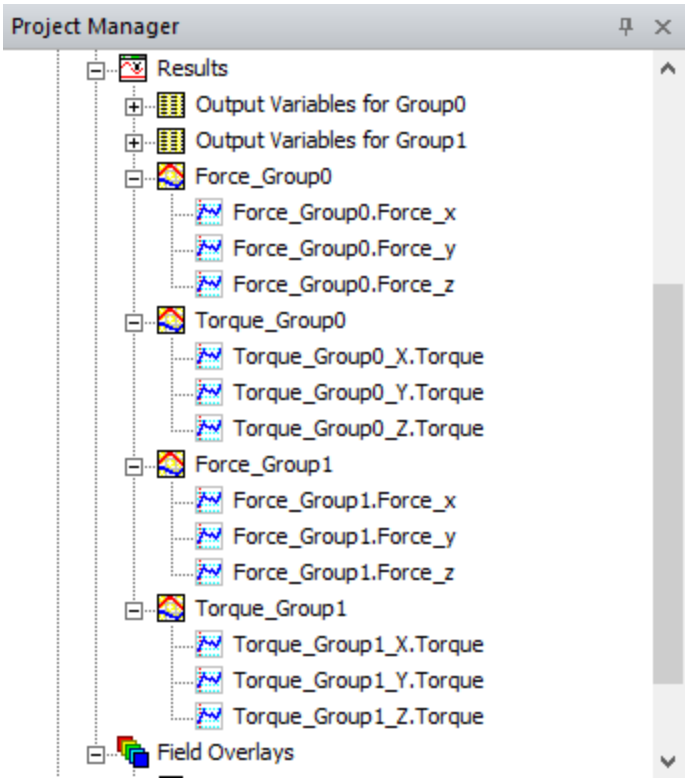


5. Click **Generate Maxwell Project**. A Maxwell desktop will be launched automatically with the generated Maxwell magnetostatic design, which includes the following:
    - The model geometry (including necessary coordinate systems)
    - If a Material is not found in the Maxwell system or user libraries, a new material with the default properties of *vacuum* will be created.
    - Force/torque calculation setups
    - A [default] Solve setup
    - Data table for force/torque parameters
    - Field plot for magnetic objects
    - Ansys Motion passes the Motion solve time step as a variable named **SimTime** to Maxwell. Excitations can be created depending on this variable.

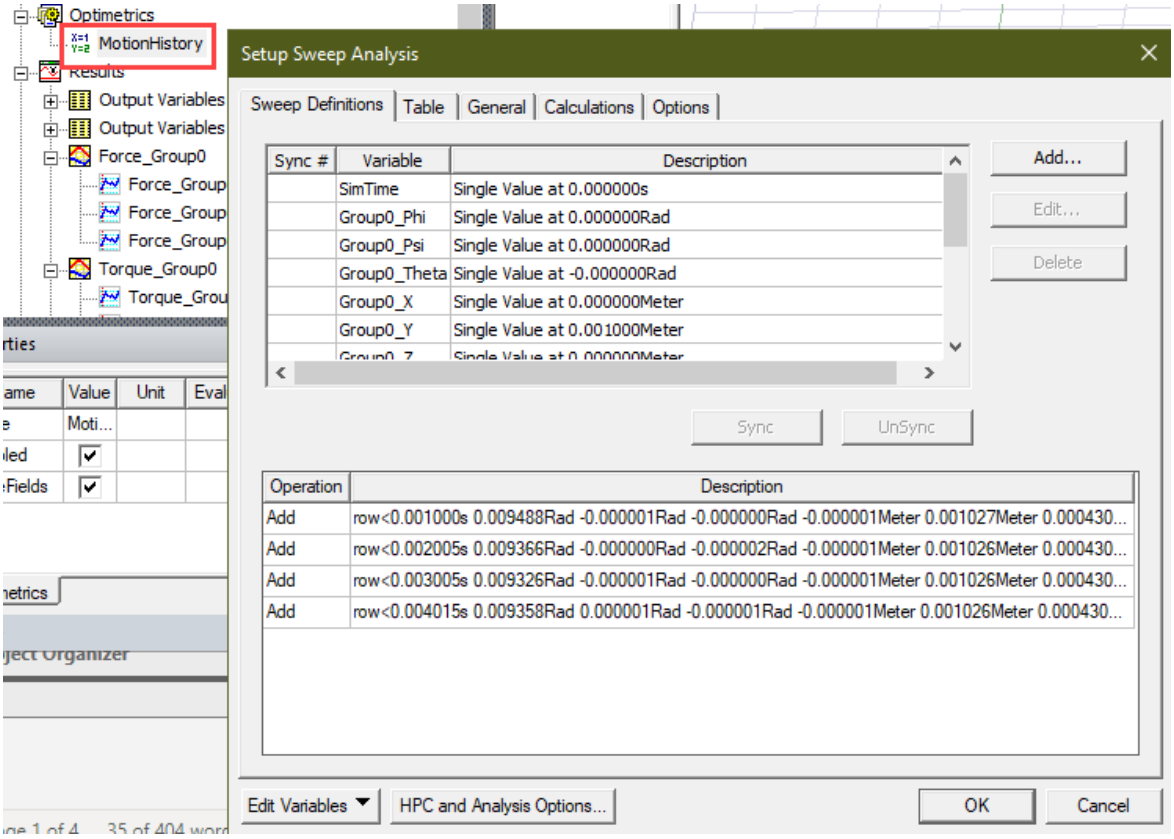


    - A parametric setup called **MotionHistory** will be automatically created and will be updated for each time step, keeping a table of variations for all the simulation steps. This data can be used to create an animation of all the simulation steps.
6. Optionally, you can customize various settings manually in Maxwell, such as:
    - Material properties for objects, especially characteristics for permanent magnets
    - Excitations can be created dependent on the **SimTime** variable.
    - Coordinate system settings
    - Mesh setup
    - Solve setup
7. After the setups on both Ansys Motion and Maxwell (if any) are completed, you can start the co-simulation solve in Motion by selecting **Simulation>Run** on the **Home** tab.

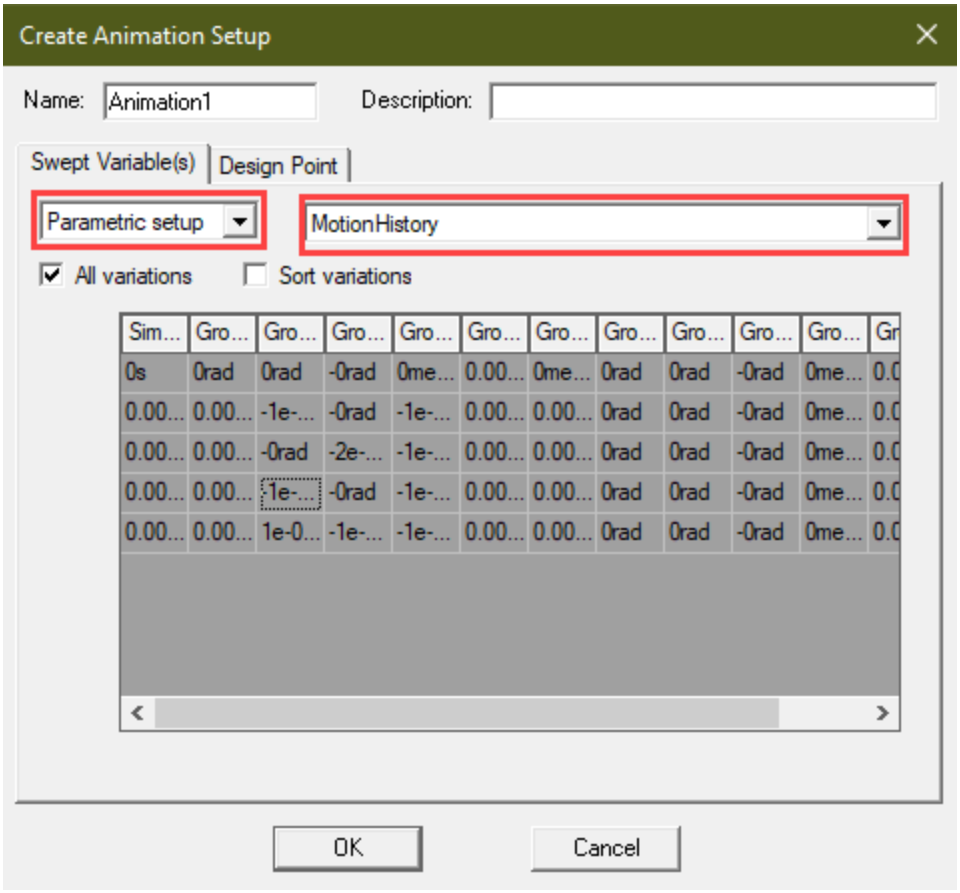
8. After co-simulation is completed, Maxwell Project Manager Results should show two reports for each group, one for force, one for torque. The reports plots are force/torque against **SimTime**.



9. Optionally, you can create an animation of all the simulation steps. A parametric setup called **MotionHistory** is automatically created and updated for each time step, keeping a table of variations for all the simulation steps.



10. Optionally, to create an animation using the automatically generated **MotionHistory** parametric setup:
    a. Right-click on the desired field plot and click **Animate…**
    b. In the **Create Animation Setup** dialog, select **Parametric setup** in the first box, then select the parametric design called **MotionHistory** in the second box.



c. Click **OK** to create the animation.

Coupling Maxwell to a Twin Builder Component

Maxwell designs may be dynamically coupled to Twin Builder components through the Twin Builder user interface. The Twin Builder interface allows Maxwell 2D, 3D, and RMxprt designs to be linked as dynamic components. Source windings set to **External** are available as conservative pins in Twin Builder.

- Transient coupling links must be enabled in Maxwell on the **Design Settings>Advanced Product Coupling** tab for Transient Cosimulation with Twin Builder.
- The ECEIM_Model component must be used in a Maxwell eddy current design to enable Twin Builder to couple to the design.

Dynamically linked components provide the ability to have Twin Builder vary a variable in a Maxwell design and have Maxwell solve the design at the new design point and provide the solution data back to Twin Builder for use in the circuit simulation.

For more information on coupling Maxwell designs to Twin Builder components, refer to the *Maxwell Coupling Components* and *RMxprt Dynamic Coupling Component* topics in the **Twin Builder Help**.

Related Topics

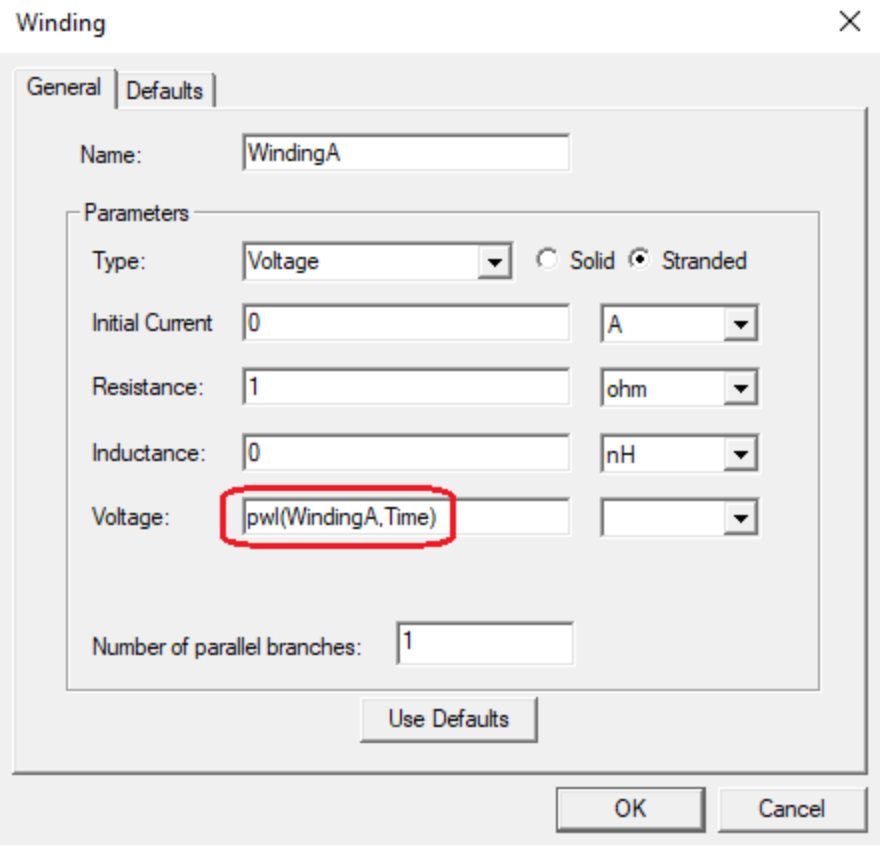
Advanced Product Coupling Tab (for Transient-Transient coupling only.)

Coupling Maxwell to a Twin Builder Excitation Component

You can couple a Maxwell 2D or 3D transient, or a 3D A-phi transient design using the Twin Builder **Excitations Component**. The Excitations component user interface allows you to assign output quantities in the Twin Builder design to windings in the coupled Maxwell design. Based on Maxwell excitations (voltage/current), Twin Builder will perform simulation, then generate detailed PWL expressions and dataset data. These expressions and data can then be pushed back to Maxwell as excitations for further analysis.

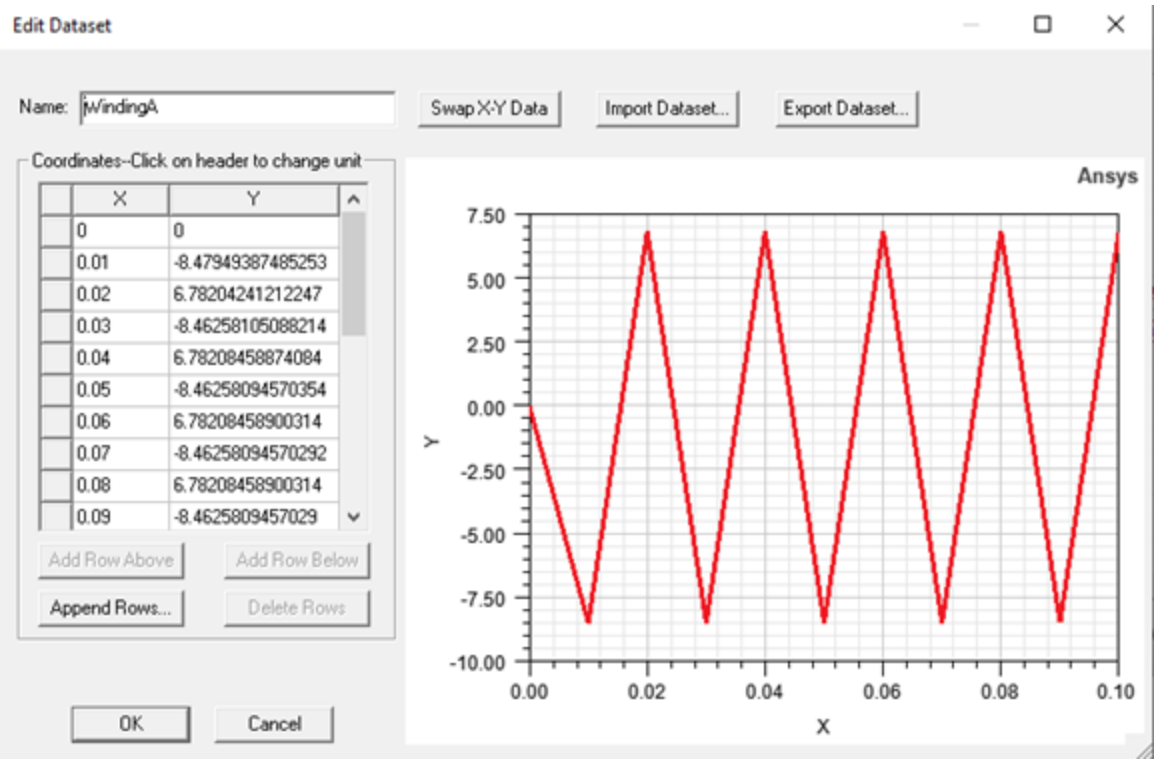
Refer to the *Maxwell Excitation Component* section in the **Twin Builder** help for the procedures on coupling a Twin Builder **Excitations Component** to a Maxwell 2D or 3D transient, or a 3D A-phi transient design; and sending **Push Excitations** data back to the coupled Maxwell design.

After the excitations push-back, you can see the push-back excitation expression in Maxwell. For example, for winding voltage:



The dataset for the PWL function can be viewed by selecting **Maxwell>Design Datasets**. In the Dataset dialog box, you can see the data set details.

To view the data set in a table, click **Edit**.



Coupling Maxwell Magnetostatic and Eddy Current Designs with Ansys Fluent

Coupling is supported between Ansys Fluent and Maxwell 3D magnetostatic and Eddy current designs. For example, a transient Fluent simulation can use a Maxwell solution to simulate the magnetic effect of the electric arc of contactors. This coupling supports designs that have a datalink defined, such as mesh, demagnetization, and so on. A design that is enabled for thermal and stress feedback is not supported.

In Ansys Fluent, during a transient simulation that requires exchanging data with Maxwell, a user-defined function (UDF) invokes the Maxwell solver. Ansys Fluent and Maxwell exchange inputs and solution data in a coupling directory that you specify. Maxwell calculates the B field based on the mesh-based conductivity (and meshed-based permeability in eddy current designs); and current excitation provided by Ansys Fluent. Ansys Fluent calculates flow field, temperature, and feedback electrical conductivity (and permeability in eddy current designs) to Maxwell.The Maxwell GUI does not provide direct access to the data in this coupling directory.

Follow these steps to set up the coupling:

1. Set up the magnetostatic or Eddy current project in Maxwell 3D.
2. In the project tree, right-click **Setup** under **Analysis** and select **Configure Fluent Conductivity Coupling**.
3. In the **Configure Fluent Conductivity Coupling** dialog box, browse to select the coupling directory or type a path, and click **OK**. You can enter a local directory, such as where Maxwell desktop is running, or you can enter a mapped network location.
   - If you enter an existing directory that contains files, a warning notifies you that all files in that directory will be permanently deleted. Click **Yes** to use that directory and delete all files contained in it. Click **No** if you do not want to select that directory. When you click No, the dialog box stays open for you to enter a different directory.
   - If the directory you entered does not exist, Maxwell creates the directory. If the directory is not successfully created, the dialog remains open.

Note	• If there are errors during the process or if you terminate the process, the coupling directory is not configured. • If there is version incompatibility between ANSYS Fluent and Maxwell, the process is stopped and an error message is displayed. • If you Clean Stop the process and there is enough solution data to configure the directory, the coupling directory will still be configured.
4. Set up the Ansys Fluent project as described in the *Ansys Fluent User's Guide*.
5. Build and load the UDF that was developed for your application. For more information about UDFs, see the *ANSYS Fluent UDF Manual*.
6. In the **input.txt file** that is specific to the UDF, provide the information that is required by the UDF. For more information about the required information and format, see Sample Input.txt File Format.

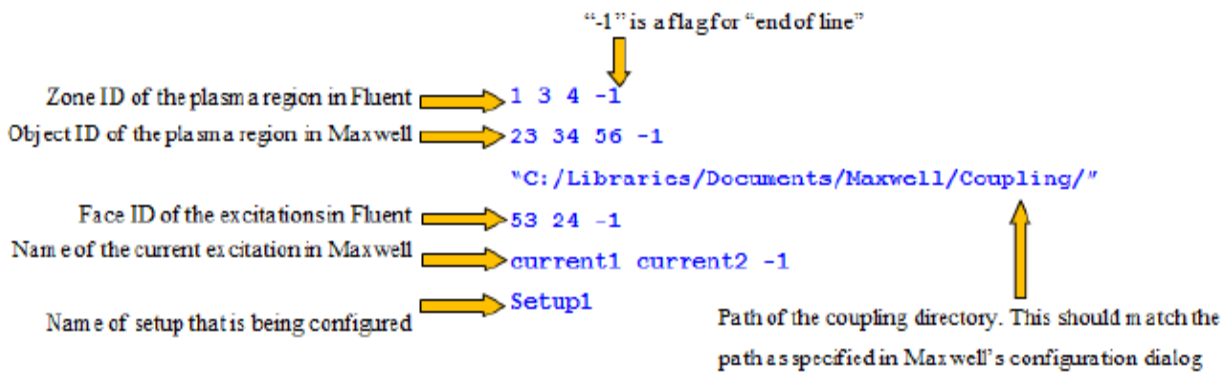
   The UDF is invoked during the transient simulation. When there is a need to exchange data, the UDF invokes the Maxwell solver.

Related Topics

Sample Input.txt File Format

Sample Input.txt File Format

The **input.txt file** is used by the UDF during the coupling between Ansys Fluent and Maxwell. The following example shows the format and provides information about the parts of the file.



Related Topics

Determining the Zone ID in Ansys Fluent
Determining the Face ID in Ansys Fluent
Determining the Object ID in Maxwell

Determining the Zone ID in Ansys Fluent

You can determine the Zone ID in Ansys Fluent as follows:

1. In the navigation pane under **Solution Setup**, select **Cell Zone Conditions**.
2. Select a zone on the task page.
   The ID of the zone you selected is displayed in the ID text box.

Related Topics

Sample Input.txt File Format

Determining the Face ID in Ansys Fluent

You can determine the Face ID in Ansys Fluent as follows:

1. In the navigation pane under **Solution Setup**, select **Boundary Conditions**.
2. Select a boundary on the task page.
   The ID of the face assignment is displayed in the ID text box.

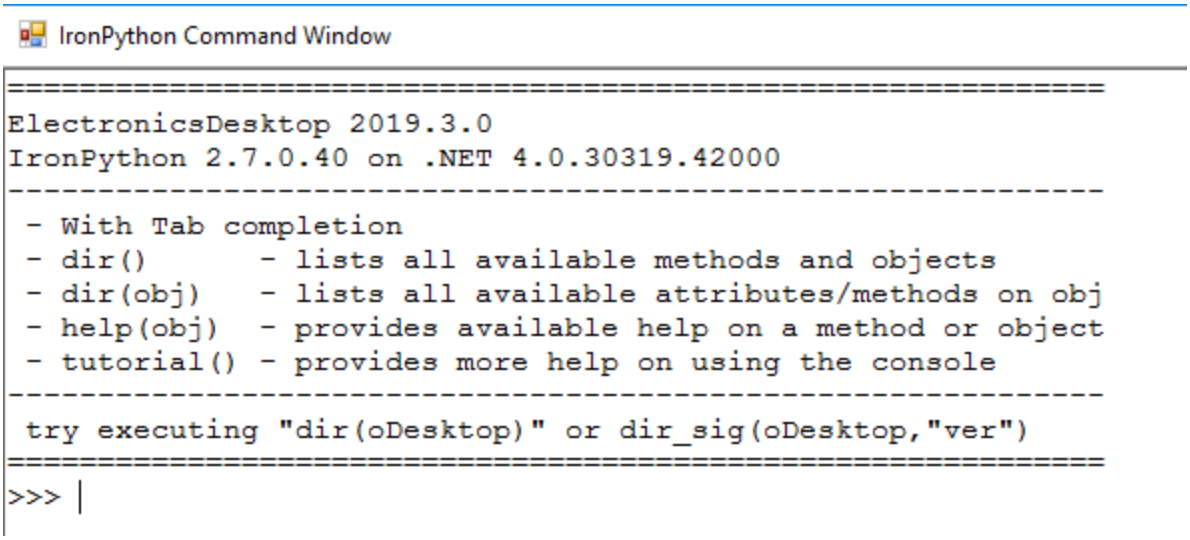
Related Topics

Sample Input.txt File Format

Determining the Object ID in Maxwell

You can determine the Object ID in Maxwell as follows:

1. Click **Tools>Open Command Window**.
2. In the **IronPython Command Window**, query the object ID by its name.



Related Topics

Sample Input.txt File Format

Magnetostriction in Maxwell Magnetostatic to Workbench Mechanical Coupling

Maxwell supports magnetostriction and inverse magnetostriction forces in Workbench Mechanical coupling for Maxwell 2D/3D Magnetosatic designs. Both one-way and two-way coupling are supported.

One-way coupling

- The Maxwell solver calculates force density which includes magnetostriction force data. The force density data including magnetostriction force data is output to the Workbench Mechanical system .

Two-way coupling

- Maxwell provides force density data to the Mechanical system as it does for one-way coupling.
- The Workbench Mechanical system solves and sends stress/strain data (plus displacement data) to Maxwell, which will be used in the Maxwell solution for the next iteration.

Related Topics

Magnetostriction and Inverse Magnetostriction

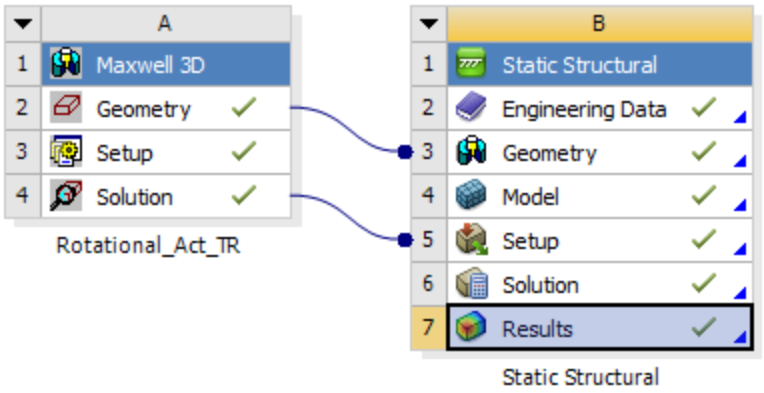
Magnetostriction Stress Coupling Between Maxwell 2D/3D Transient and Workbench Mechanical

Maxwell supports magnetostriction forces in two-way coupling between Workbench Mechanical and Maxwell 2D/3D Transient designs.

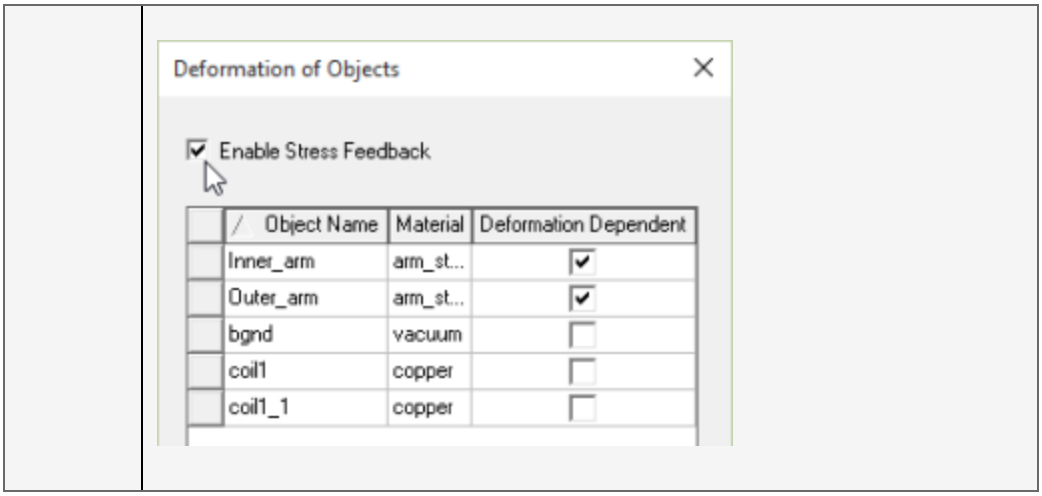
- Maxwell 2D/3D transient provides force density data to the Mechanical system.
- The Workbench Mechanical system solves and sends stress/strain data (plus displacement data) to Maxwell, which will be used in the Maxwell solution for the next iteration.

To analyze the effect of stress due to magnetostriction forces on objects in a design using two-way coupling between a Maxwell 2D/3D transient design and Workbench Mechanical:

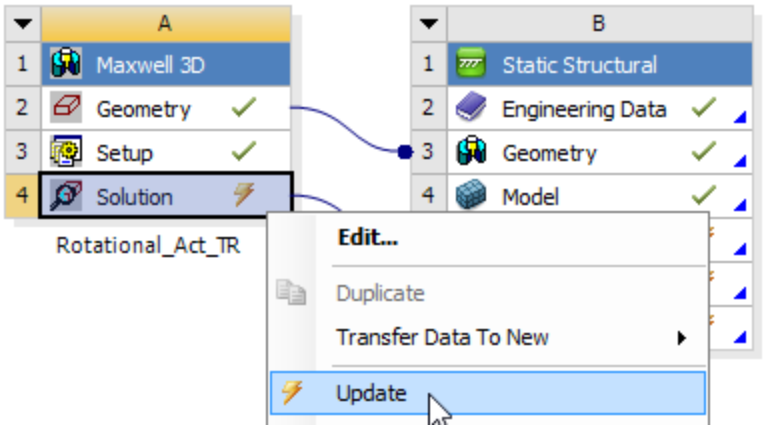
1. Set up the Maxwell3D/2D transient Mechanical structure coupling.



Note	You must manually Enable Stress Feedback and select the desired stress feedback objects in the Maxwell design.

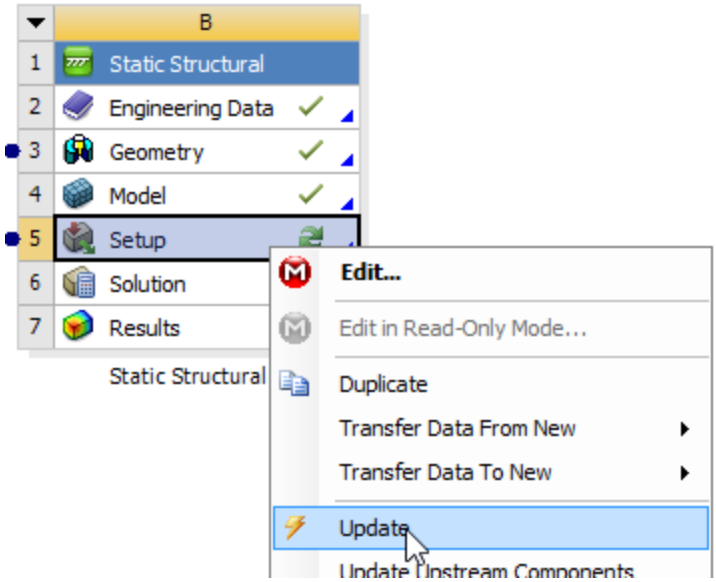


2. **Update** the Maxwell **Solution** cell to initiate a Maxwell solve.

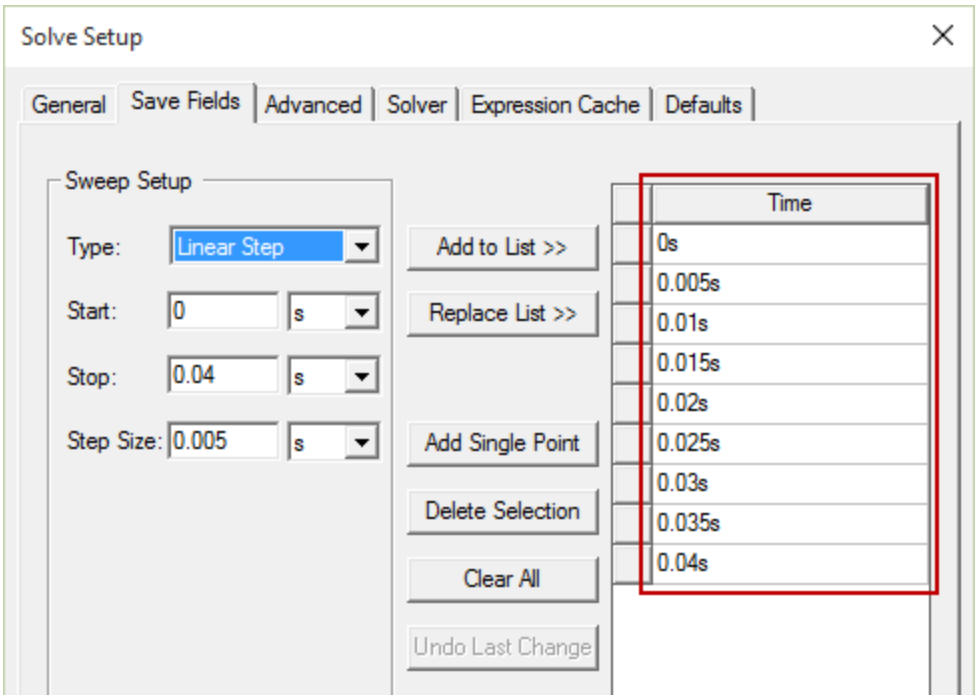


The Maxwell transient solver generates force density data, which can be used by Workbench Mechanical.

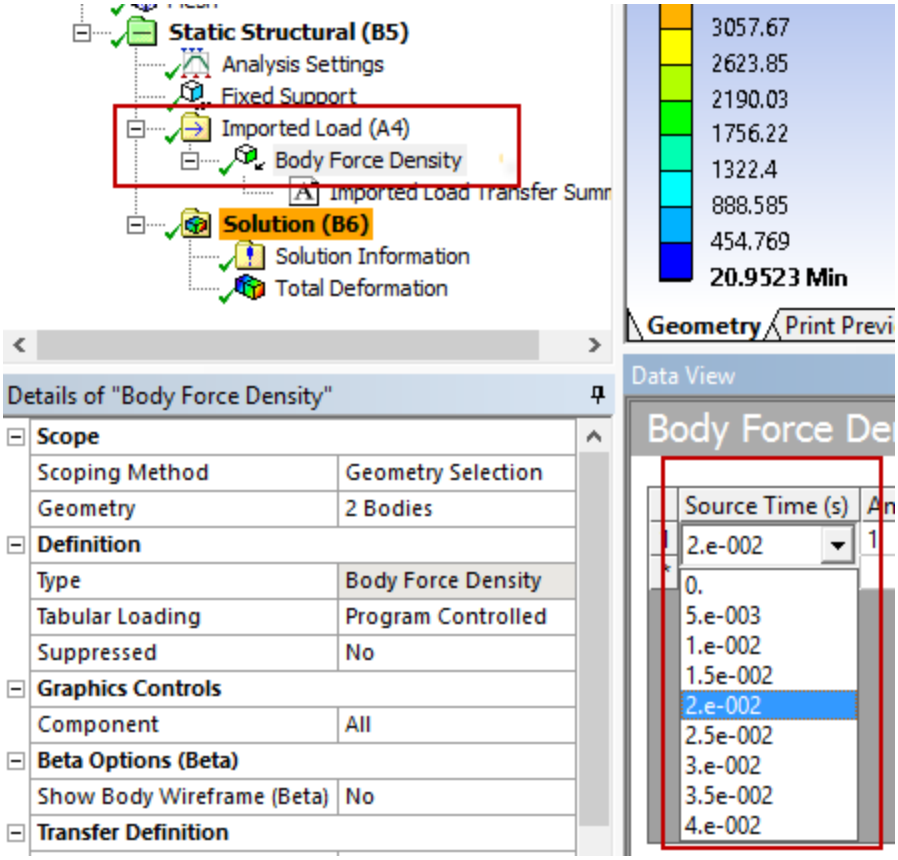
3. **Update** the Workbench Mechanical **Setup** cell to import the force density data generated by Maxwell.



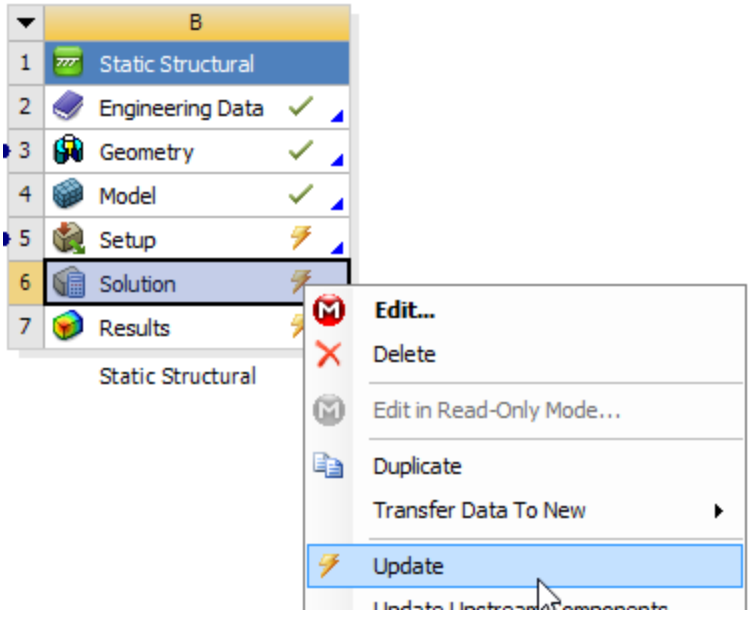
Maxwell maps the Maxwell mesh to the Workbench Mechanical mesh; and the calculated force density values for the time points you set up on the Maxwell transient **Solve Setup** dialog **Save Fields** tab are loaded into Workbench Mechanical.



4. In Workbench Mechanical, select the desired source time point you want to use for the force density calculations.



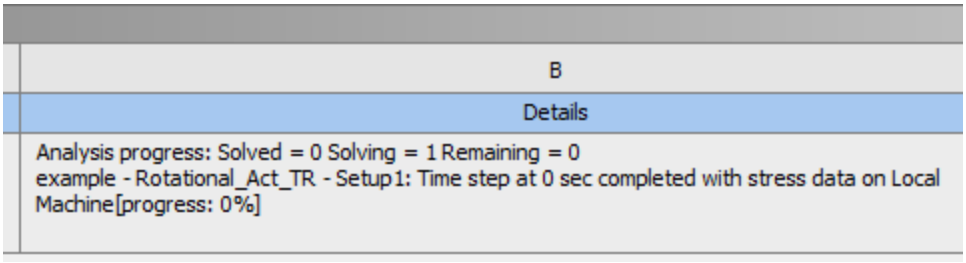
5. **Update** the **Solution** cell of the Mechanical panel.



Workbench Mechanical solves the structural project using the force density at the selected source time, then outputs displacement and stress/strain data to be used as feedback by the Maxwell transient solver.

6. To re-solve using the feedback data, set the **Solution** cell of the Maxwell panel to **Enable Update**, then perform an **Update** for the Solution cell. (You can also use the feedback iterator to automatically update the Maxwell Solution cell.)

   The Maxwell 3D/2D transient solver uses the feedback displacement and stress/strain data to re-solve from time 0.



Related Topics

Magnetostriction and Inverse Magnetostriction

Setting Deformation of Objects for Stress Feedback

18 - Ansys Workbench Integration Overview

Ansys Workbench combines the strength of its core product solvers with the project management tools necessary to manage project workflow. In Ansys Workbench, analyses are built as *systems*, which can then be combined into a *project*. The project is driven by a schematic workflow that manages the connections between the systems.

From the schematic, you can interact with applications (called workspaces) that are native to Ansys Workbench and that display within the Ansys Workbench interface. Native workspaces include: Project Schematic, Engineering Data, and Design Exploration (Parameters and Design Points).

You can also launch applications that are data-integrated with Ansys Workbench, meaning the application's interface remains separate, but the data from the application communicates with the native ANSYS Workbench data. Thus, data can be passed back and forth between any Ansys Electromagnetics product on a Workbench Project Schematic and any supported Ansys or Ansys Electromagnetics desktop product. Depending on the application, data integration can include basic actions such as saving projects, as well as more complex actions such as the coupling of Ansys Electromagnetics product variables to Workbench Design Exploration parameters.

Data-integrated applications include the following Ansys Electromagnetics products: Circuit, HFSS, Maxwell/RMxprt, Q3D Extractor, and Twin Builder.

Note	For detailed information on working with Ansys Workbench, please refer to the Workbench documentation.

Integrating Ansys Electromagnetics Products with Ansys Workbench

Workbench Data Integration Overview

Ansys Electromagnetics - Ansys Multiphysics Coupling

Ansys Electromagnetics CAD Integration Through Workbench

Workbench Material Data Transfer

Ansys Electromagnetics to Ansys Geometry Transfer

User Defined Model (UDM) for Ansys WB Integration

Feedback Iterator

Integrating Ansys Electromagnetics Suite Products with Ansys Workbench

You can integrate Ansys Electromagnetics products with Ansys Workbench in one of two ways:

- During product installation, or
- After product installation

Integration with Ansys Workbench during Ansys Electromagnetics Product Installation

You can choose to integrate Ansys Electromagnetics products with Ansys 2023 R1 during installation as follows:

1. If you have not already installed Ansys 2023 R1, do so before proceeding. Refer to the Ansys documentation for instructions on installing the Ansys 2023 R1 software.
2. Launch **Autorun.exe** or **Setup.exe** for the Ansys Electromagnetics product you wish to install.
3. Proceed through the installation dialogs.
4. When the **Integration with Ansys 2023 R1** dialog displays, select the **Yes** radio button to have the installer automatically integrate the Ansys Electromagnetics product with Ansys 2023 R1.

Note	If you choose not to integrate the product with Ansys 2023 R1 during installation, you can perform this step after installation.

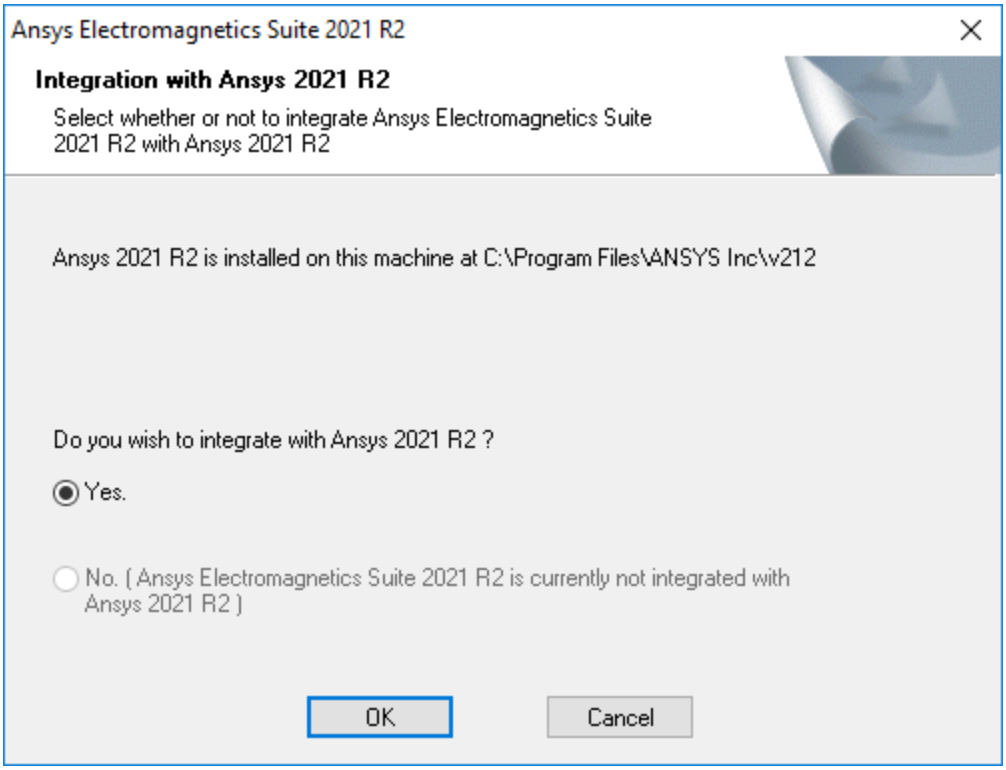
4. Complete the product installation and exit the installer.

Integration with Ansys Workbench after Ansys Electromagnetics Product Installation

If you want to integrate Ansys Electromagnetics Suite with Ansys 2023 R1 after installation, do the following:

1. From the **Start** menu, select **Ansys EM Suite 2023 R1** > **Modify Integration with Ansys 2023 R1**.
2. In the **Integration with Ansys 2023 R1** dialog box, ensure that the **Yes** radio button is selected, then click **OK** to complete the integration process. Then **OK** the information

dialog.



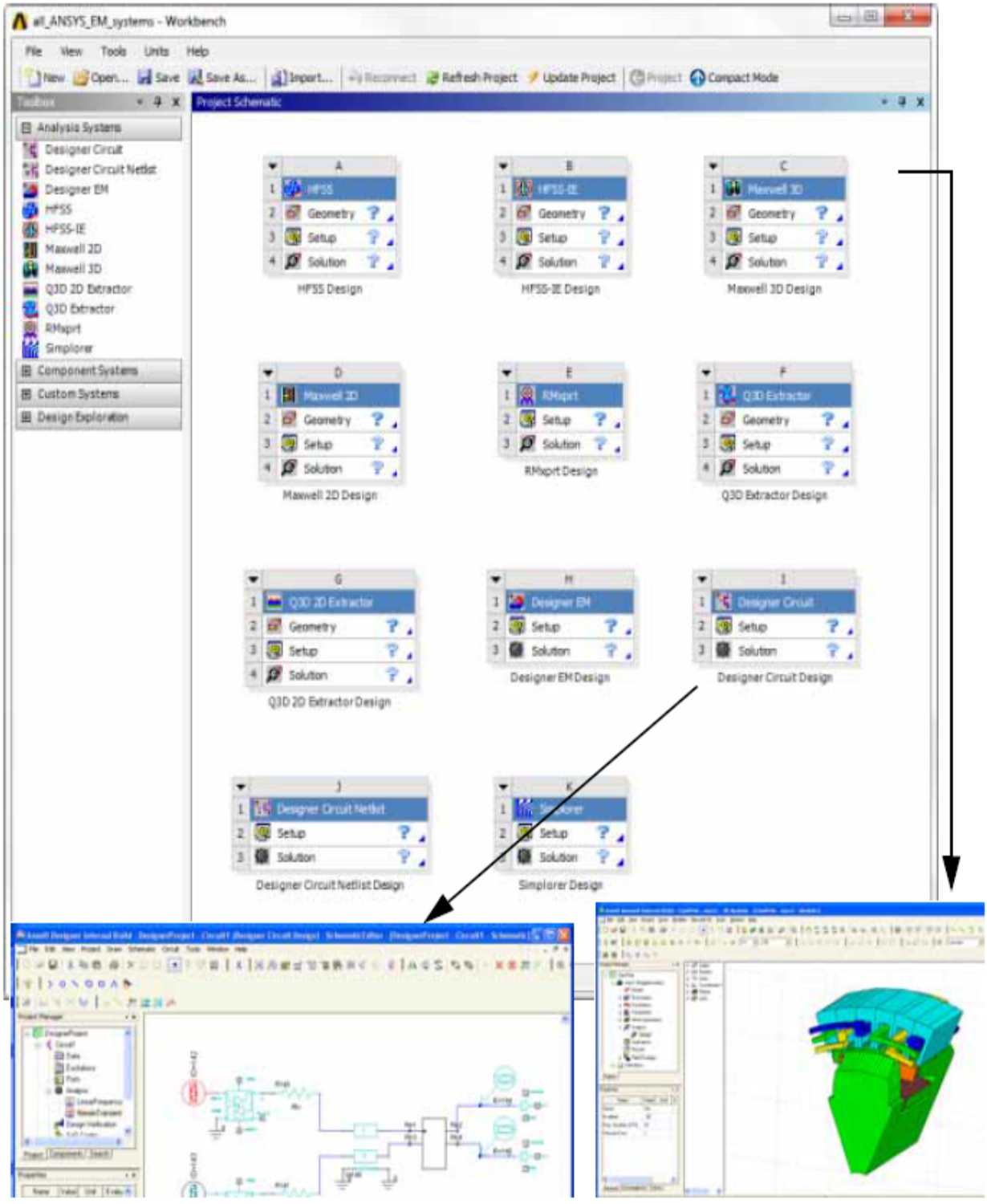
You can confirm that Ansys 2023 R1 is “aware” of the Ansys Electromagnetics Electronics Desktop application via the Ansys Workbench Options dialog box, which shows the Electronics Desktop install path.

The Journal Recording Language option permits you specify the default journal recording language.

- The default language is IronPython. You can change the recording language to VB.
- Changes in the recording language option apply to any Ansys Electromagnetics desktop launched after the change.
- Any open Ansys Electromagnetics desktop will continue recording in the language that was in effect when it was launched.

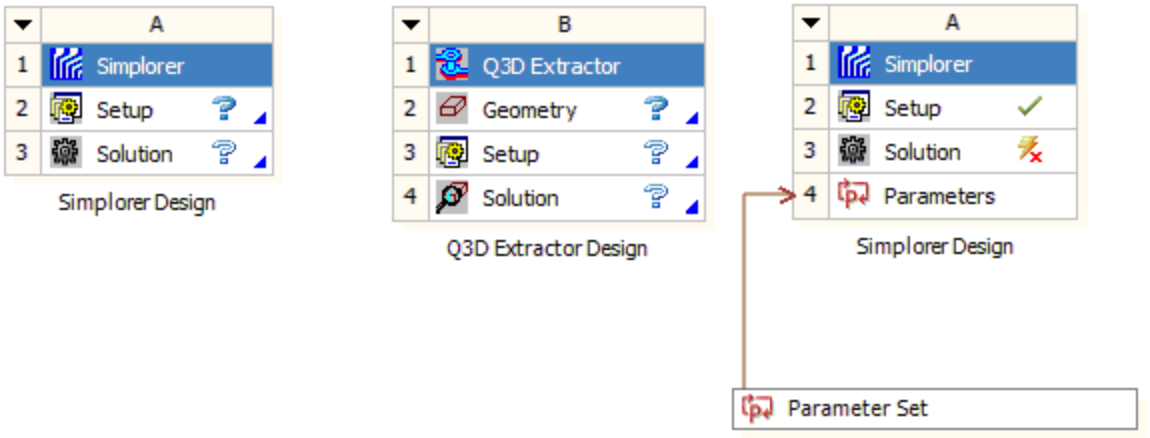
Workbench Data Integration Overview

Ansys Electromagnetics data-integrated applications can reside on a Workbench **Project Schematic** as shown below.



Objects, such as instances of Ansys Electromagnetics projects, that are placed on a Workbench Project Schematic are referred to as *systems*. Ansys Electromagnetics circuit/system products: RMxprt, Circuit, and Simplorer, appear on Workbench Project Schematics as systems with two

"cells" – **Setup** and **Solution**. Ansys Electromagnetics field products: HFSS, Maxwell, and Q3D Extractor add an additional **Geometry** cell. If you invoke Ansys **DesignXplorer** to use variables for refining a design, a **Parameters** cell is added with a link to the associated Workbench **Parameter Set**. Refer to the Ansys 2023 R1 Workbench help for details on working with systems, cells, and parameter sets.



Ansys Electromagnetics desktop products integrate with Workbench commands, services, and DesignXplorer in a similar manner. Here are some of the ways in which Ansys Electromagnetics products integrate with Workbench:

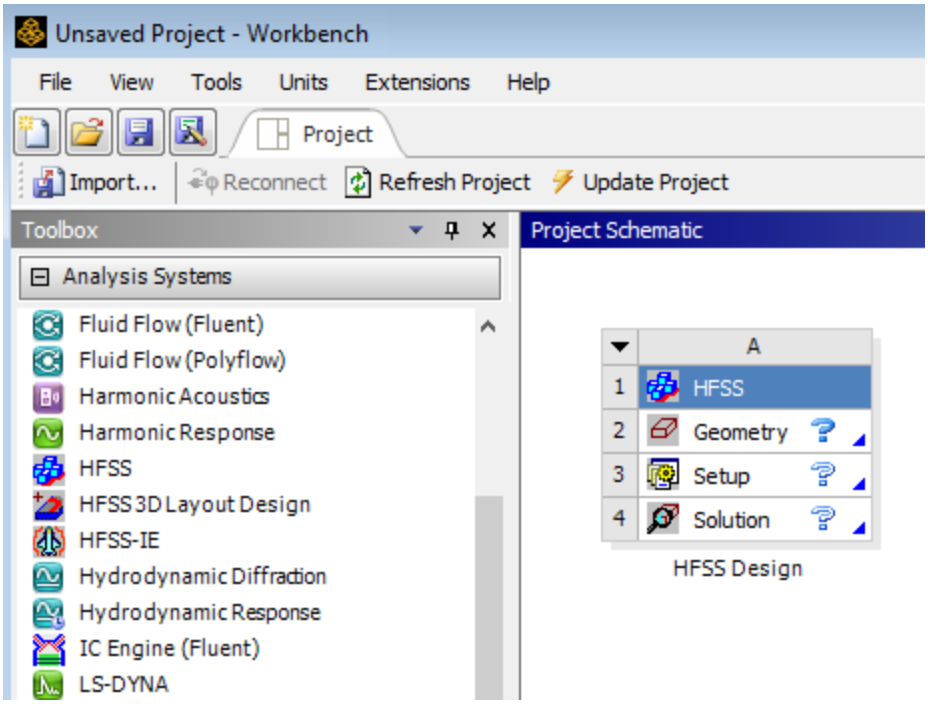
- Adding new analysis systems
- Importing existing desktop projects
- Editing models
- Analyzing models
- Performing parameter studies
- Scripting

In addition to these major features, Workbench also allows you to Archive, Save, Backup, Duplicate, and Delete Ansys Electromagnetics projects used in a Workbench project. Progress information and messages from integrated Ansys Electromagnetics projects are also displayed in Workbench.

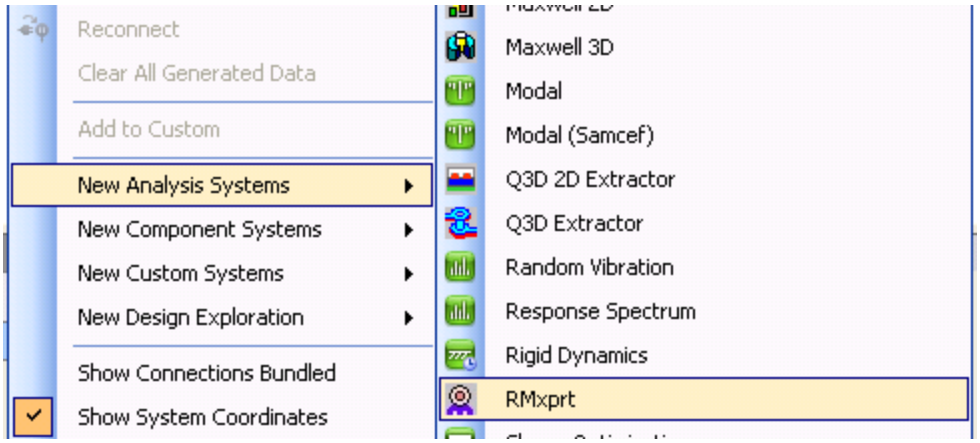
Note	Detailed information for using these operations can be found in the Ansys Workbench help.

Adding New Ansys Electromagnetics Analysis Systems

A new Ansys Electromagnetics Analysis System can be added to a Workbench Project Schematic either by dragging and dropping it from the Toolbox:



or by selecting it from the context menu in the Workbench Project Schematic window:



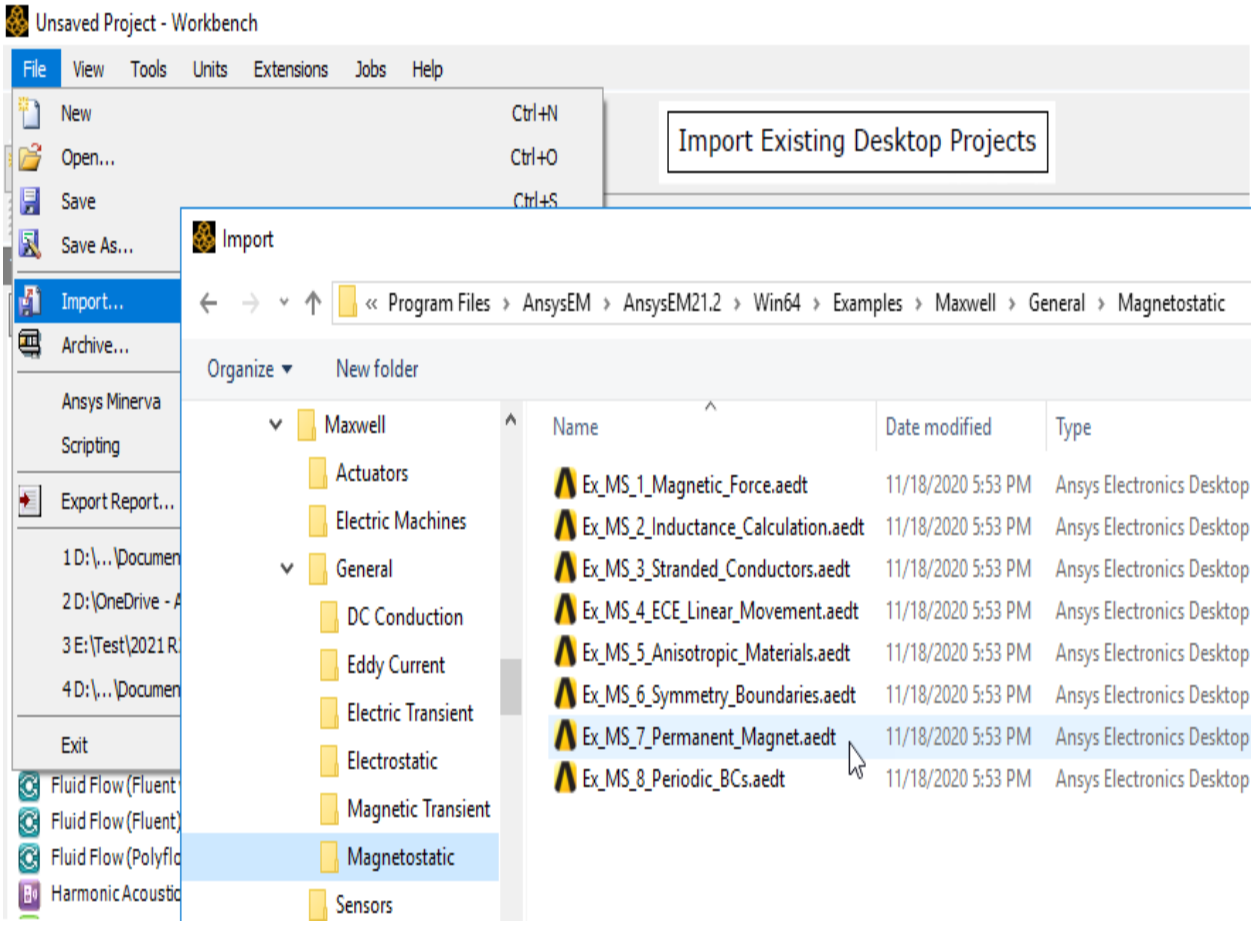
Related Topics

Workbench Data Integration Overview
Importing Ansys Electromagnetics Projects into Ansys Workbench
Editing Ansys Electromagnetics Models in Workbench
Analyzing Ansys Electromagnetics Models in Workbench
Performing Parameter Studies in Workbench
Scripting in Workbench

Importing Ansys Electromagnetics Projects into Ansys Workbench

You can import existing Ansys Electromagnetics desktop projects to a Workbench Project Schematic. When you do, a copy of the Ansys Electromagnetics project is put into the Workbench Project folder. The original Ansys Electromagnetics project remains intact.

Note	Object, material, and parameter names with non-ASCII characters are not allowed for data transfer. Such transfers fail and produce an error message.



Related Topics

Workbench Data Integration Overview
Adding New Ansys Electromagnetics Analysis Systems
Editing Ansys Electromagnetics Models in Workbench
Analyzing Ansys Electromagnetics Models in Workbench
Performing Parameter Studies in Workbench
Scripting in Workbench

Editing Ansys Electromagnetics Models in Workbench

You can edit various properties and parameters (geometry, setup, solution, etc.) of the Ansys Electromagnetics project either by right-clicking on the project in Workbench and selecting **Edit** on the context menu; or by double-clicking the project. Doing so launches the Ansys Electromagnetics desktop application and loads the project so that you can setup your project in a familiar desktop environment. Changes made to the Ansys Electromagnetics project are saved to the project instance in the Workbench project folder.

Related Topics

Workbench Data Integration Overview

Adding New Ansys Electromagnetics Analysis Systems

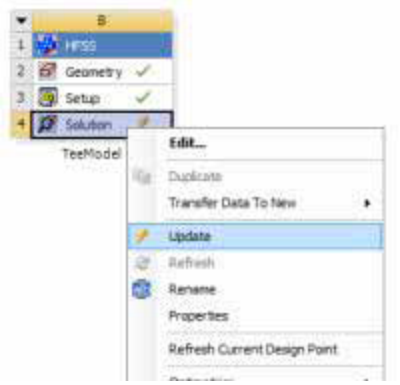
Importing Ansys Electromagnetics Projects into Ansys Workbench

Analyzing Ansys Electromagnetics Models in Workbench

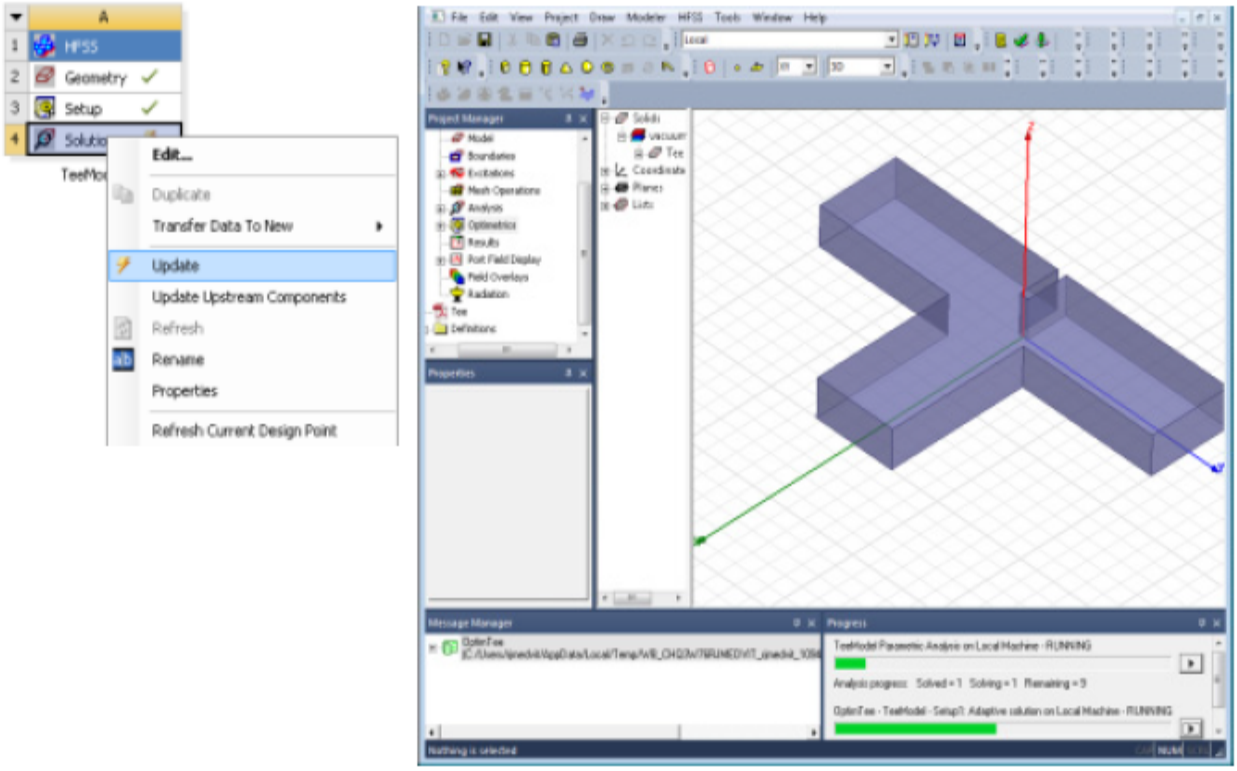
Performing Parameter Studies in Workbench

Scripting in Workbench

Analyzing Ansys Electromagnetics Models in Workbench



You can use Workbench's **Update** command to run analyses in the integrated Ansys Electromagnetics project. Progress information is also shown in Workbench.



Related Topics

Workbench Data Integration Overview

Adding New Ansys Electromagnetics Analysis Systems

Importing Ansys Electromagnetics Projects into Ansys Workbench

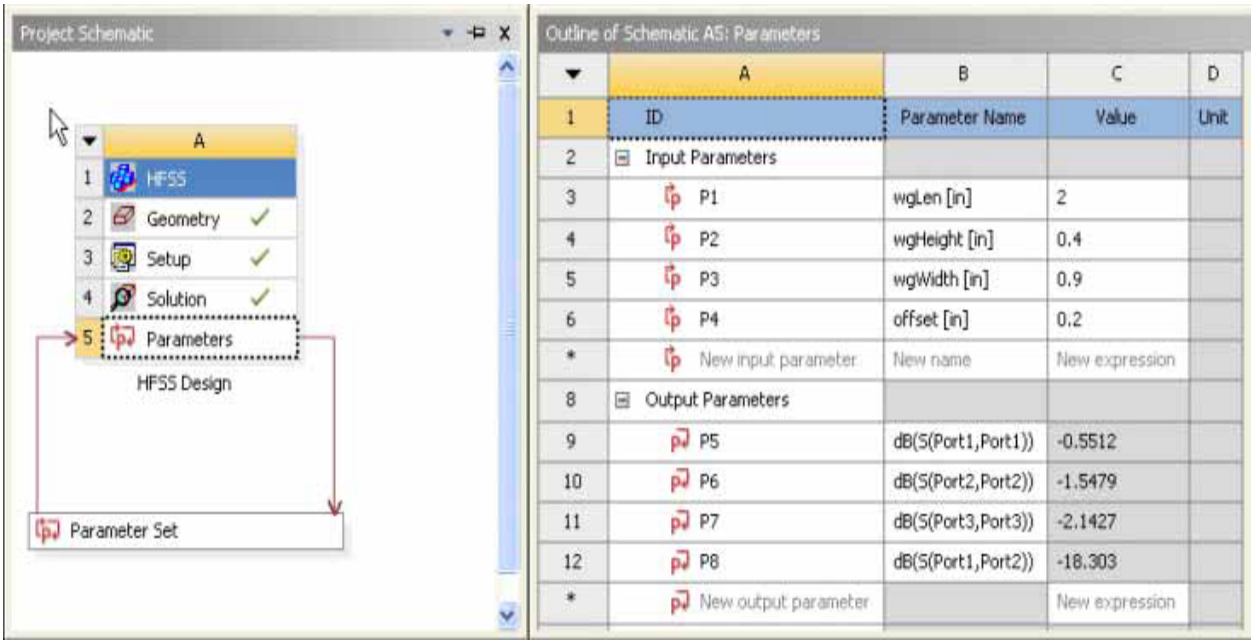
Editing Ansys Electromagnetics Models in Workbench

Performing Parameter Studies in Workbench

Scripting in Workbench

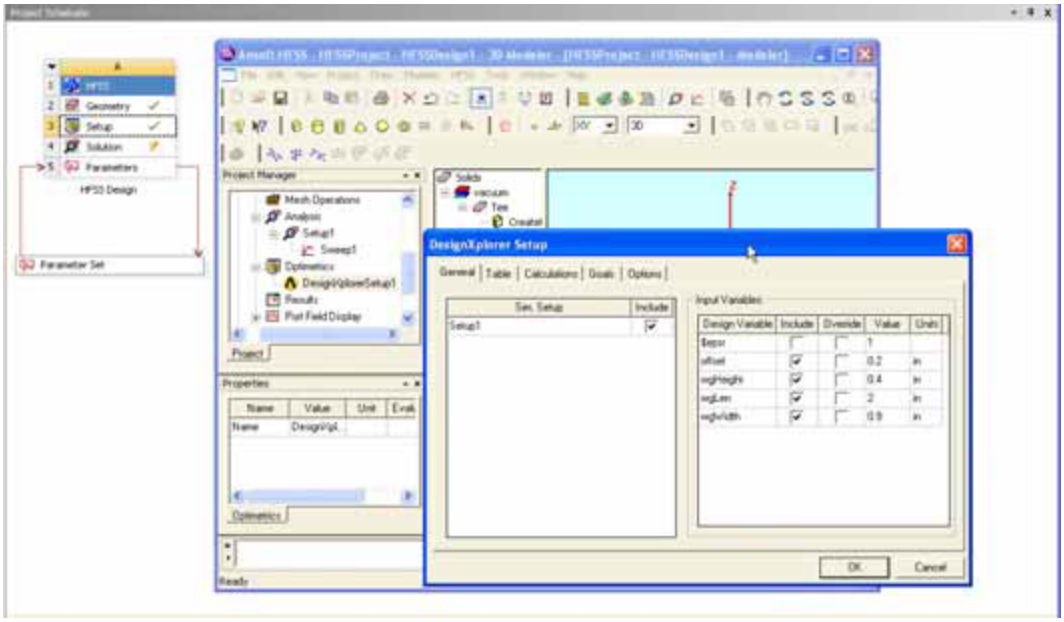
Performing Parameter Studies in Workbench

Workbench **Parameter Sets** allow you to change parameter values and units of measure, or add new parameters. Parameter data is passed back to the Ansys Electromagnetics application for updated analyses. After analysis, the Workbench Parameter table should be updated correctly for all design points.



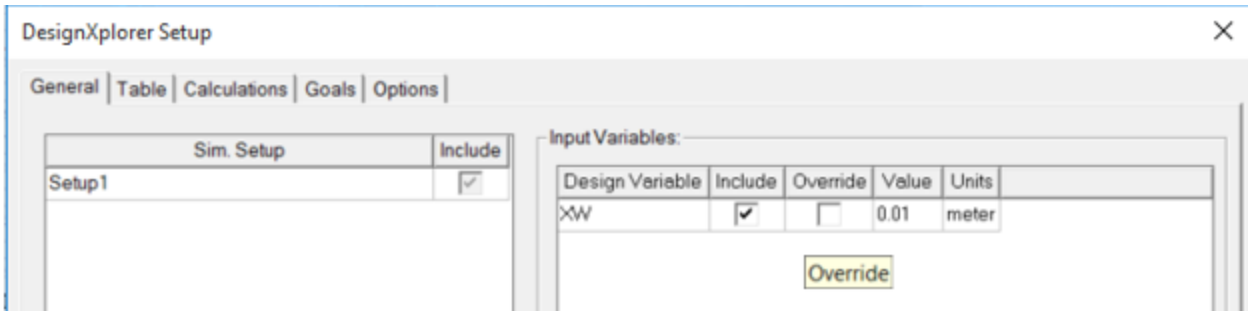
Note	Because Workbench and Ansys Electromagnetics Desktop have different unit systems, units defined in Desktop are Dimensionless in the Workbench Parameters table and thus cannot be changed. Refer to the *ANSYS Design Modeler User's Guide* > *Project Schematic Operations* > *Parameters in Project Schematic* > *Parameter Units* topic for additional information.

Parameters from the Ansys Electromagnetics project are exposed to Workbench through the **DesignXplorer** setup. The Ansys Electromagnetics system's cell status on the Workbench project is updated as changes are made in the Ansys Electromagnetics application desktop.

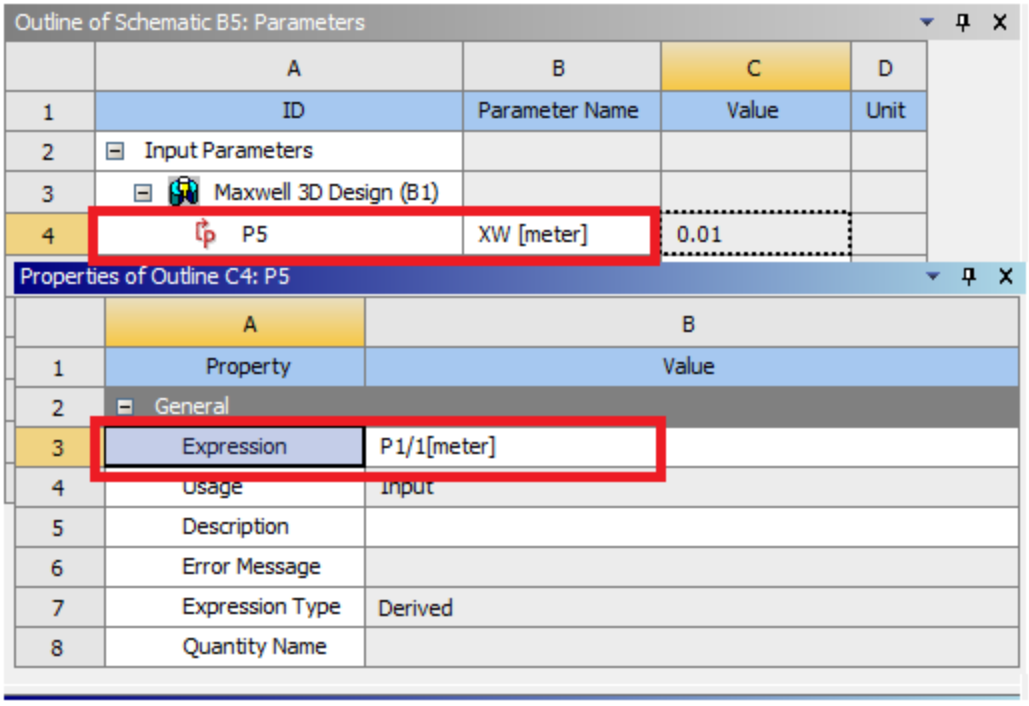


The workflow for using Electronics Desktop systems with the Workbench to take advantage of Distributed Analysis is as follows.

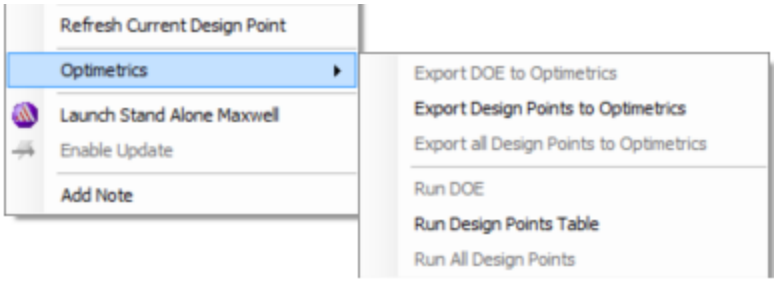
1. In the Ansys Electronics Desktop, specify a variable an as input parameter. The following figure shows a DesignXplorer Setup that includes a geometry variable called XW This variable is mapped to a variable in the corresponding Workbench Design, in this example, P5.



2. In the Workbench Design, change the P5 expression to a new value (in this example figure, the P5 parameter is assigned and expression of P1/1[meter] to tie it to the geometry system's variable.



In the Workbench, the **Export Design Points to Optimetrics** and **Run Design Points Table** commands are enabled.



3. In the Electronics Desktop, you can click on the default DesignXplorer Setup, and choose **Generate Variation Data**.

   If you have configured your Analysis Configuration for multiple machines, the Project is ready for Distributed Analysis. After analysis, the Workbench Parameter table should be updated correctly for all design points.

Related Topics

Workbench Data Integration Overview

Adding New Ansys Electromagnetics Analysis Systems

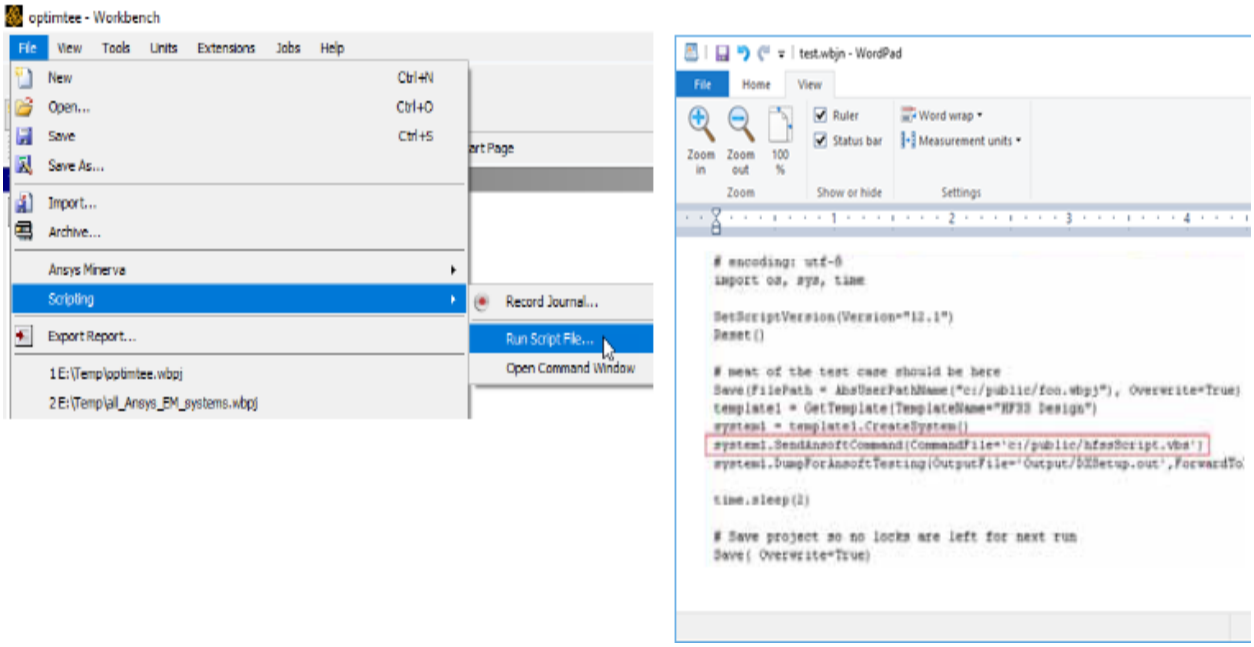
Importing Ansys Electromagnetics Projects into Ansys Workbench

Editing Ansys Electromagnetics Models in Workbench

Analyzing Ansys Electromagnetics Models in Workbench
Scripting in Workbench

Scripting in Workbench

Scripts that include Ansys Electromagnetics projects can be recorded and played back via Workbench.



When a WB journal is run from the command line, a beta feature allows the AnsysEDT portion to also run in batch mode: i.e, when using RunWB2 -B (In prior releases, a significant portion of recorded journal would error out requesting interactive mode playback). To use this beta feature, set the following environment variable before running the script.

From a Windows Command prompt, for instance,

```
Set ANSYSEM_FEATURE_SF6694_NON_GRAPHICAL_COMMAND_EXECUTION_
ENABLE=1
RunWB2 -B -R myReplayJournal.wbjn
```

Related Topics

Workbench Data Integration Overview
Adding New Ansys Electromagnetics Analysis Systems
Importing Ansys Electromagnetics Projects into Ansys Workbench
Editing Ansys Electromagnetics Models in Workbench
Analyzing Ansys Electromagnetics Models in Workbench
Performing Parameter Studies in Workbench

Ansys Electromagnetics - Ansys Multiphysics Coupling

Data integration provides improved multiphysics workflow between Ansys Electromagnetics designs and Ansys applications such as Mechanical and Thermal. Coupling is provided through project schematic links. Heat losses and force data are automatically transferred to Ansys Mechanical - there is no need to export/import “transfer xml files”. Edits you make in Ansys Electromagnetics applications are automatically transferred to the Ansys application through a Workbench **Refresh** command. Workbench commands also enable easier automation of iterative coupling of thermal feedback. The following sections provide some examples of multiphysics coupling.

Multiphysics Coupling on Workbench with Ansys Thermal

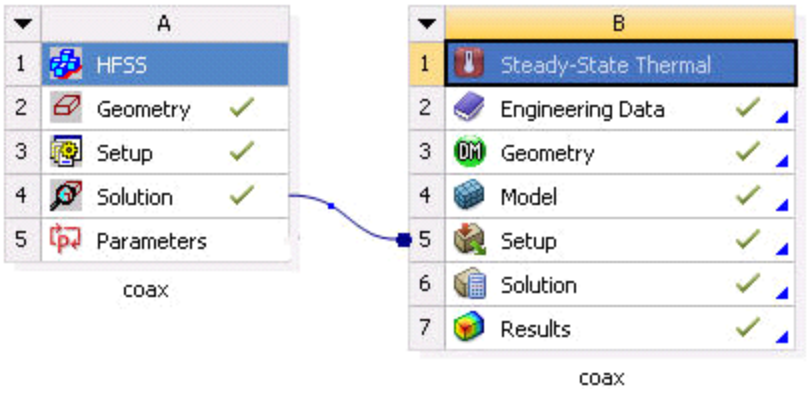
Multiphysics Coupling on Workbench with Ansys Structural

Multiphysics Coupling between Ansys Electromagnetics Field Systems on Workbench

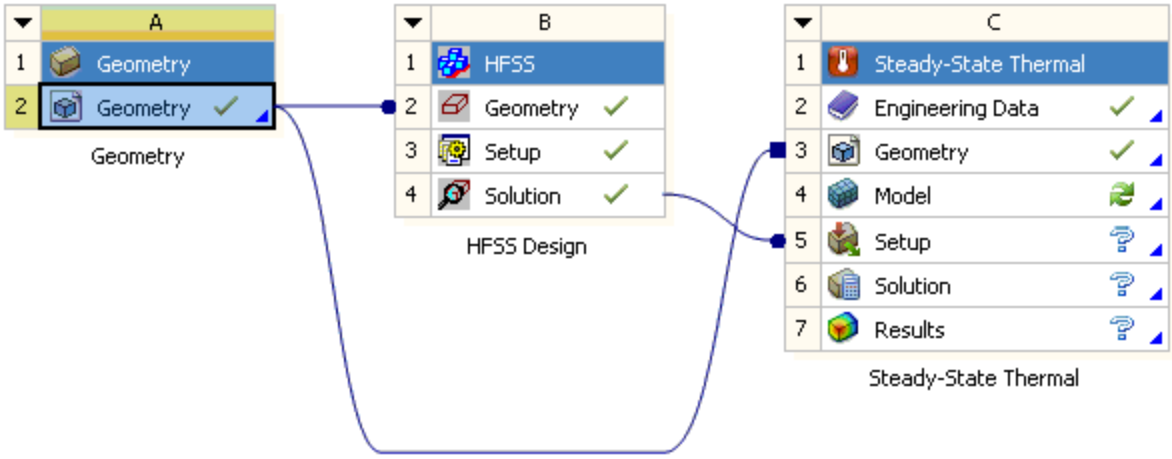
Multiphysics Coupling on Workbench with ANSYS Thermal

Using data integration, HFSS, Maxwell, and Q3D Extractor provide heat losses (heat generation and heat flux) to Ansys Thermal. You first need to enable feedback as described in Setting the Temperature of Objects.

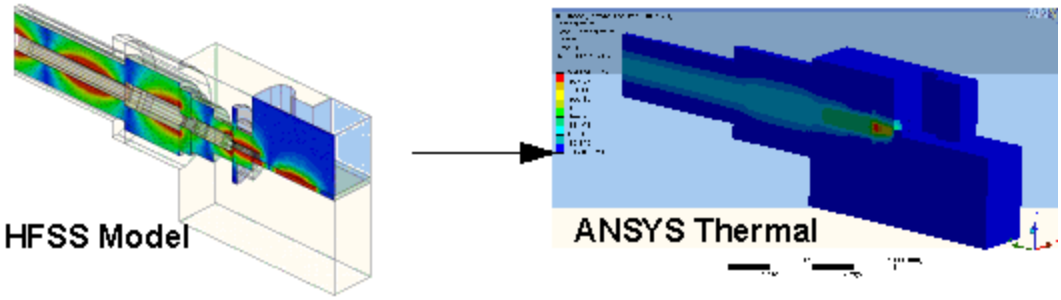
Note how the HFSS design is linked visually to ANSYS Thermal on the Workbench project schematic.



A geometry from the Component Systems in the Toolbox in Workbench Schematic can be shared as shown in the image below.



In this example, HFSS coax model Solution provides heat loss data as a thermal load to the Ansys Thermal Setup. The resulting analysis shows a thermal “hotspot”, providing the user with the information needed to adjust the design’s material to fix the problem.



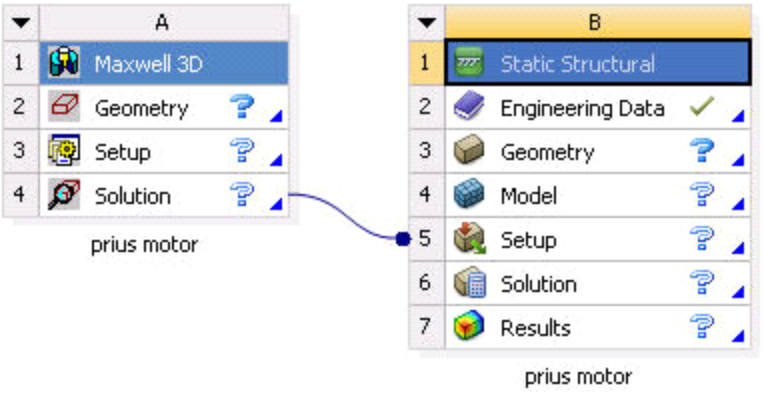
You enable this feature by checking Enable Feedback in the dialog for Setting the Temperature of Objects.

Related Topics

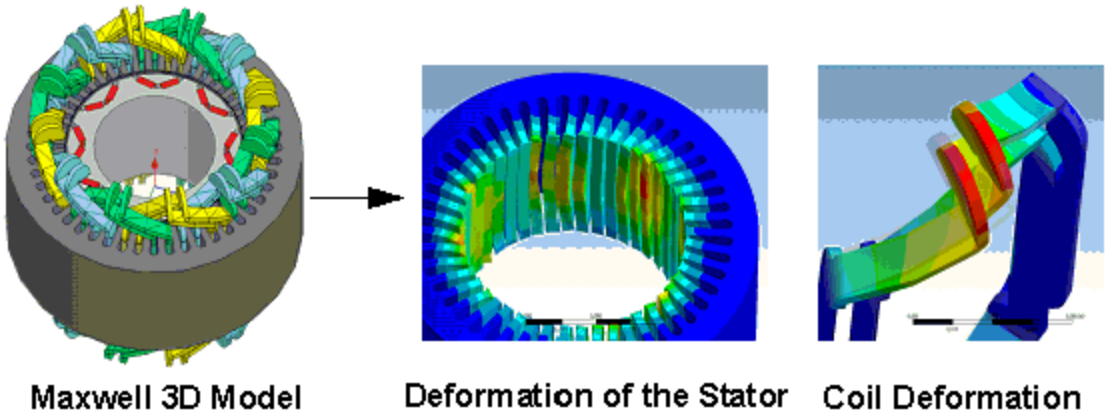
Feedback Iterator

Multiphysics Coupling on Workbench with Ansys Structural

Using data integration, Maxwell 2D and Maxwell 3Dcan provide forces to Ansys Structural.

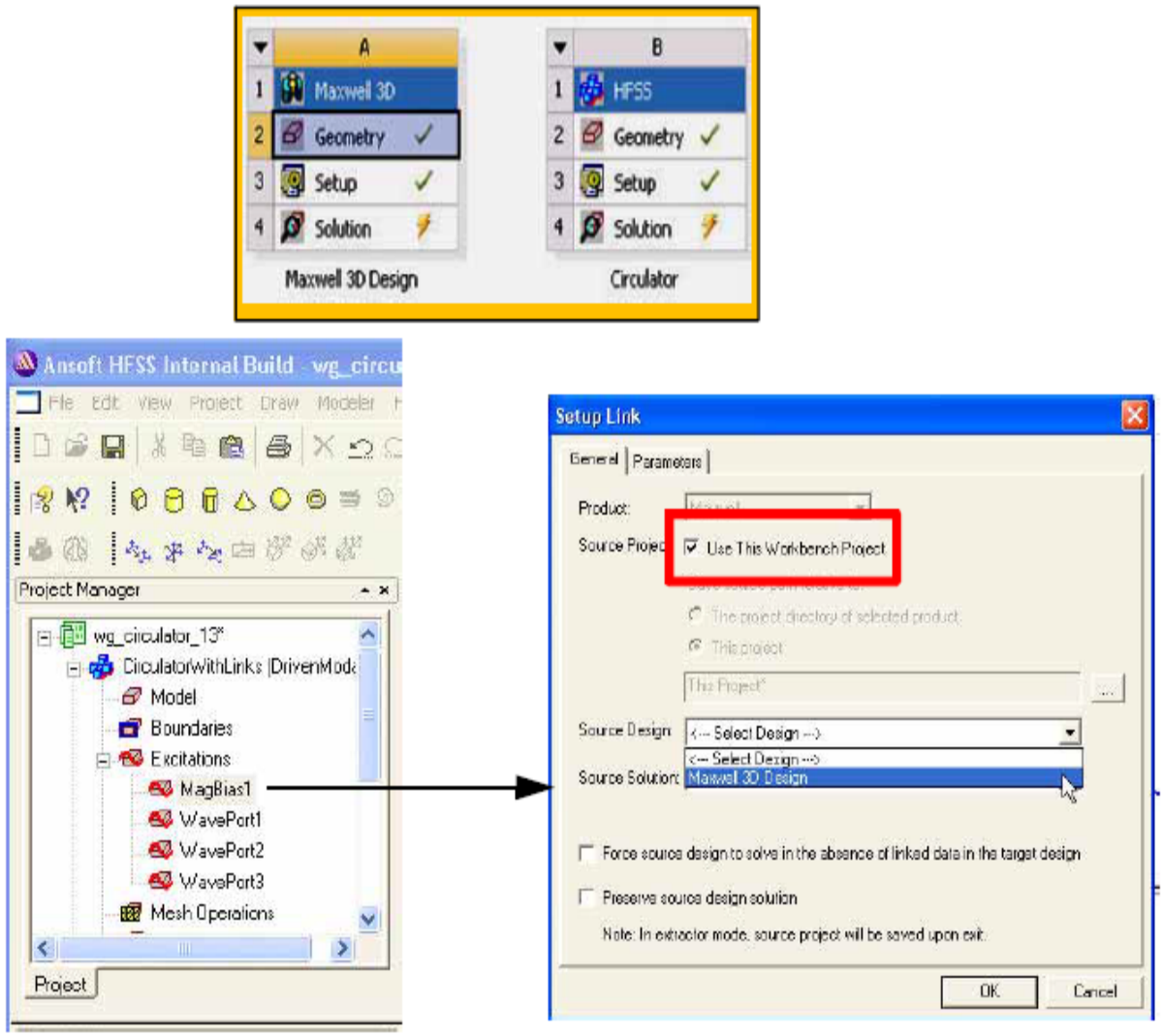


In this example, the Maxwell 3D electromagnetic force density **Solution** is used as the load in Ansys Structural to determine how these forces deform the motor's stator and coils.



Multiphysics Coupling between Ansys Electromagnetics Field Systems on Workbench

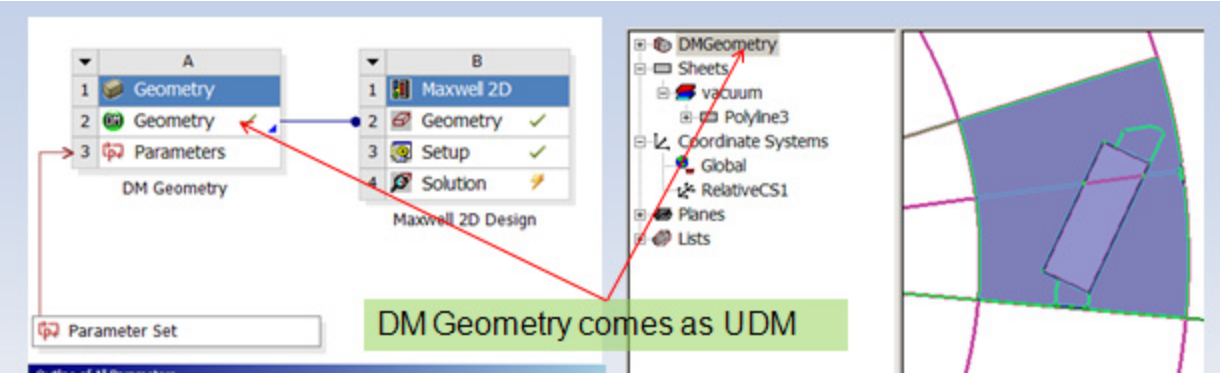
You can setup links between Ansys Electromagnetics field systems that reside on a Workbench project schematic. Linking is setup in the Ansys Electromagnetics application as shown in the following example.



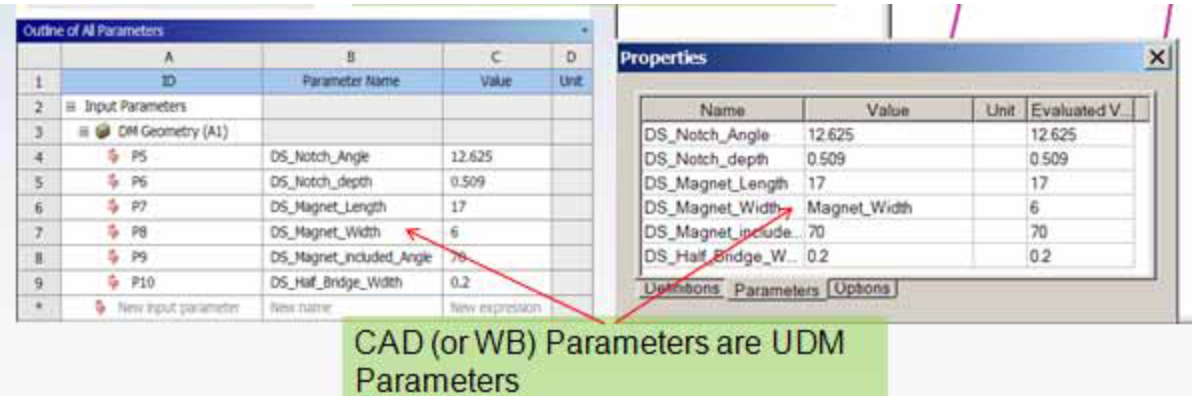
CAD Integration through Workbench

Ansys Electromagnetics CAD integration is a Workbench feature available for Ansys Electromagnetics 3D Products - HFSS, Maxwell, and Q3D as the Ansys Framework for Ansys Electromagnetics package. This feature is available only through Workbench, and is not available in standalone Ansys Electromagnetics products.

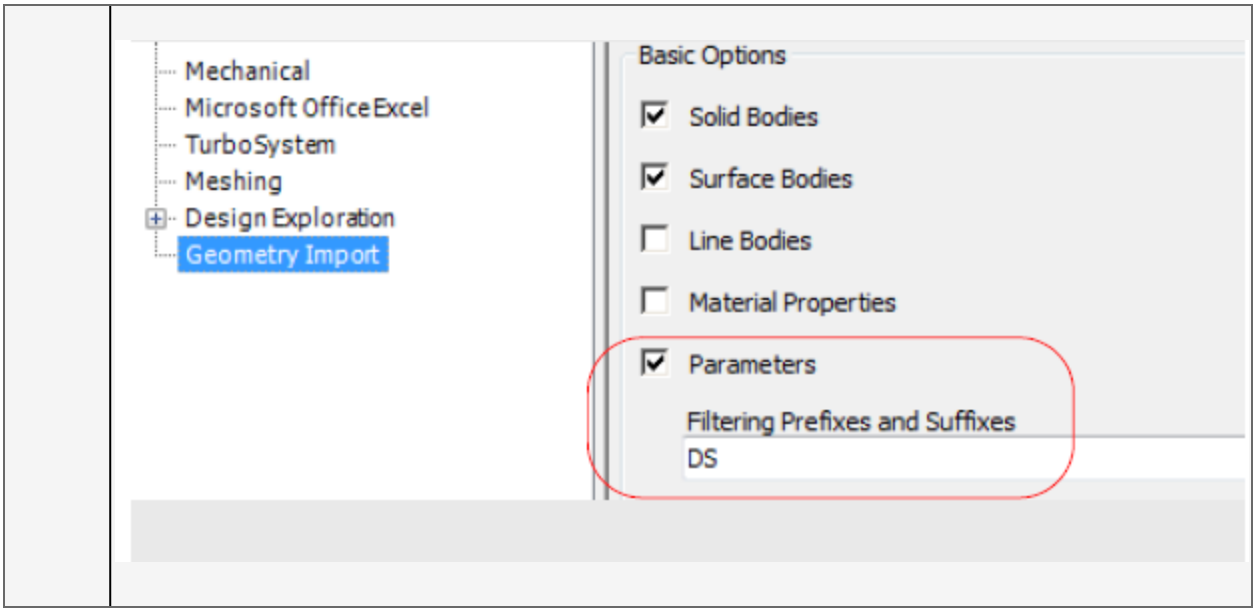
Ansys Electromagnetics CAD integration provides a bi-directional dynamic link through Workbench, which makes it possible to get updated geometry from CAD, to modify the CAD parameters in Ansys Electromagnetics products, and return updated geometry. The feature is non-associative due to a need to reassign boundaries if the modified CAD model is to be used. The process creates a User Defined Model (UDM) for each geometry source



The User Defined Model (UDM) format makes it possible to exchange parameters.



Note	The parameters shown in the previous example all have a **DS** prefix. This is the default for the Workbench **Tools>Options** for **Geometry Import.** If you want to import parameters with different prefixes or names, you should assign an appropriate prefix, or clear the Filtering Prefixes field, depending on your needs.



See here for further description of the UDM feature and function.

Ansys Electromagnetics CAD integration makes it possible to consume geometry from multiple upstream source which can be any CAD or Ansys Electromagnetics product. This feature supports direct interfaces with all major CAD systems.

- Creo Parametric
- UG NX
- CATIA V5
- SOLIDWORKS
- Autodesk Inventor
- Ansys Design Modeler (DM)
- Ansys SpaceClaim Direct Modeler (SCDM)

CAD software must be installed on user machine

- Not required on solve nodes

Platforms Supported

- Windows 64 bit
- Linux

See the following sections:

CAD Integration and Geometry Sharing

Bi-Directional CAD Integration

CAD Integration Model Edits

Multiple Geometry Links for CAD Integration

CAD Integration Functionality

Healing with CAD Integration

Important Geometry Options for CAD Integration

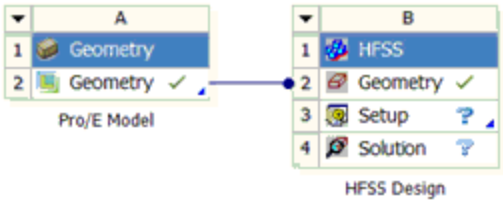
CAD Integration and Geometry Sharing

CAD model comes into Ansys Electromagnetics as User Defined Model (UDM).

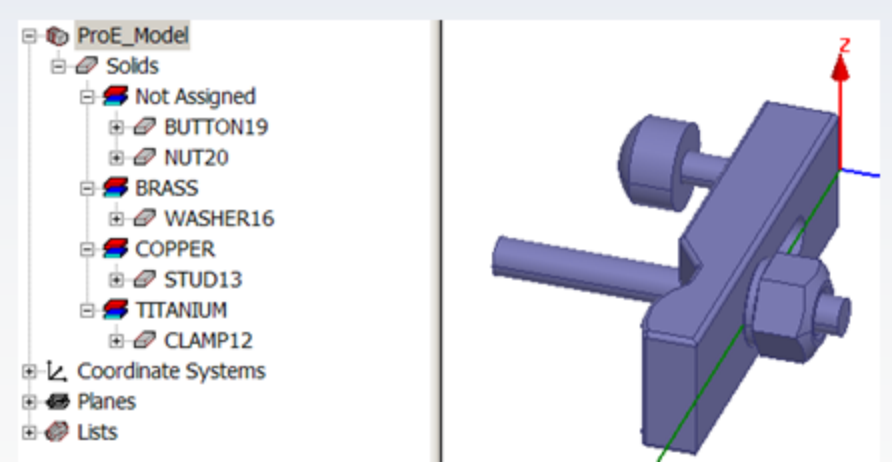
The input to Ansys Electromagnetics from CAD is:

- Geometry/Topology with persistent IDs
- CAD parameters
- Material assignment
- Attributes like name, and color

For example, in Workbench, a Pro/E Model can be linked to HFSS.



The geometry can then be viewed in HFSS as a UDM.

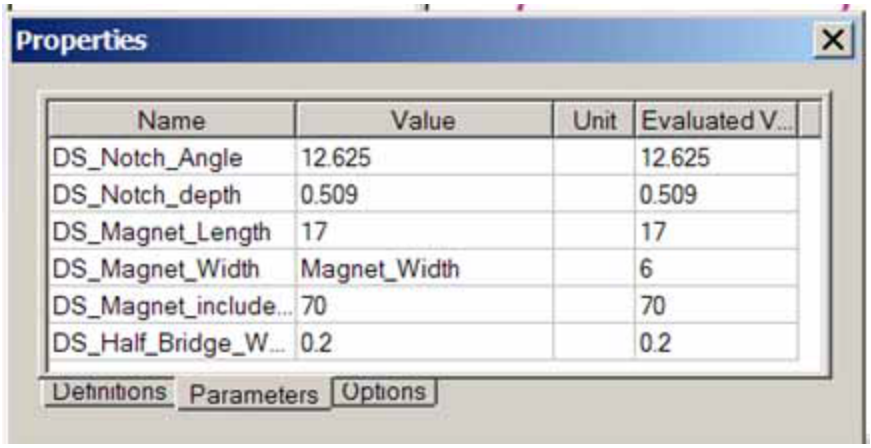


The CAD or WB model parameters appear in the Workbench:

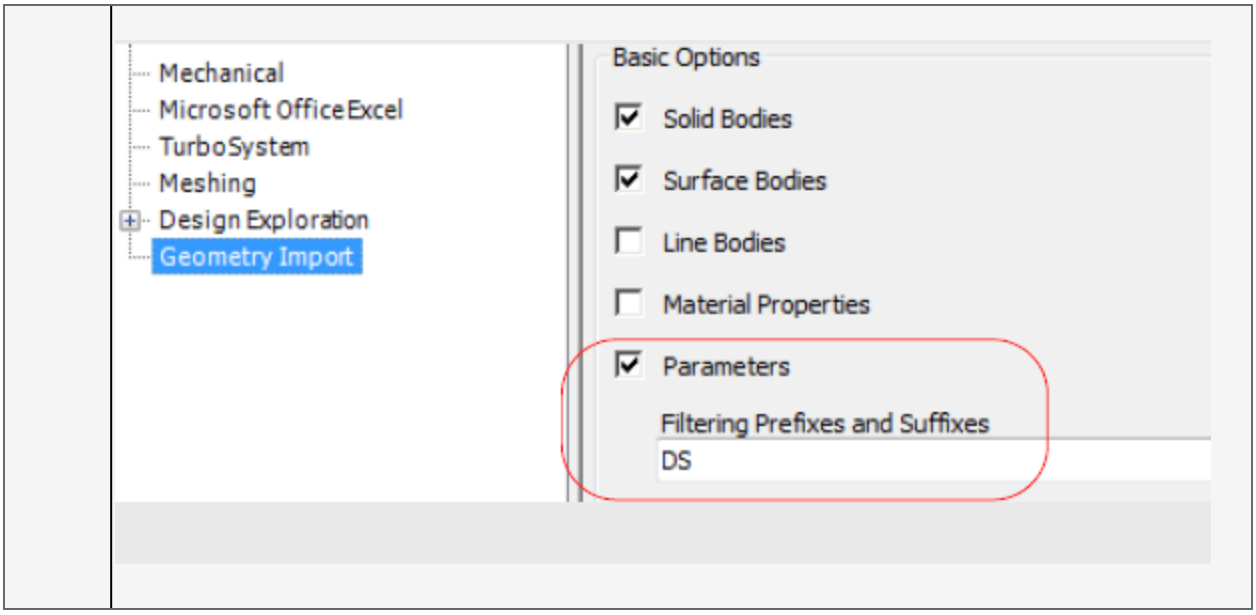
Outline of All Parameters

	A	B	C	D
1	ID	Parameter Name	Value	Unit
2	Input Parameters			
3	DM Geometry (A1)			
4	P5	DS_Notch_Angle	12.625	
5	P6	DS_Notch_depth	0.509	
6	P7	DS_Magnet_Length	17	
7	P8	DS_Magnet_Width	6	
8	P9	DS_Magnet_included_Angle	70	
9	P10	DS_Half_Bridge_Width	0.2	
*	New input parameter	New name	New expression	

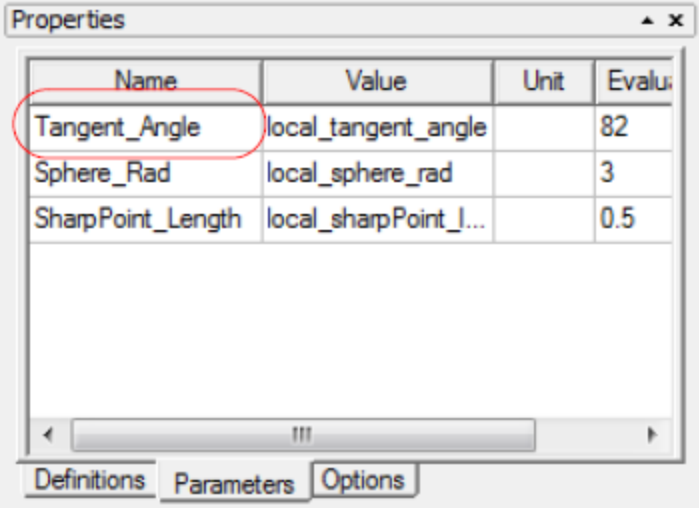
Though the Ansys Electromagnetics CAD Integration, the linked UDM includes the same parameters.



Note	The parameters shown in these examples all have a “DS” prefix. This is the default for the Workbench **Tools>Options** for **Geometry Import.** If you want to import parameters with different prefixes or names, you should assign an appropriate prefix, or clear the Filtering Prefixes field, depending on your needs.



Once you import a geometry with parameters to an Electromagnetics application in the desktop, you need to map them to local variables. That is, in the target Electromagnetics application, select the geometry associated with the parameters. Update the geometry, and view the Parameters tab in the Property window. The Values column will show the values of the imported parameters. You then type names for local variables in the values column.



Once you have local variables for the imported Design Modeler parameters, you are ready to use Design Xplorer for multiple design parameters.

Table of Schematic D2: Design of Experiments (Custom)

	A	B	C
1	Name	P3 - Tangent_Angle (degree)	P4 - Sphere_Rad (in)
2	1	80	3
3	2	75	3
4	3	85	3

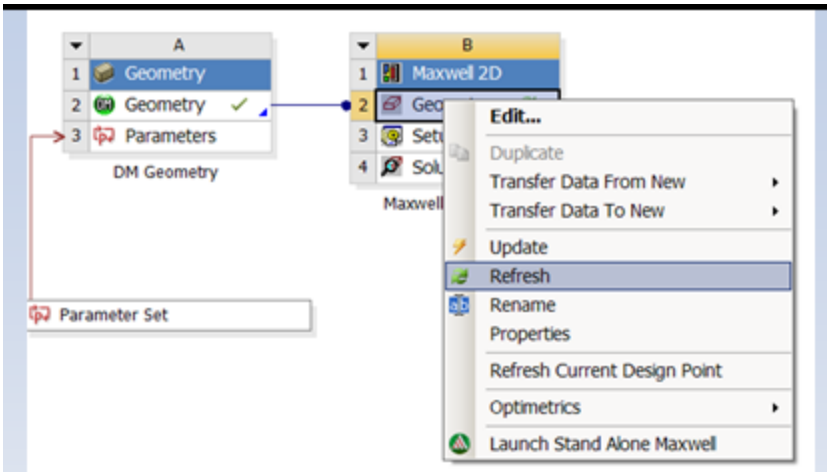
Related Topics

Ansys Electromagnetics CAD Integration Through Workbench
Bi-Directional CAD Integration
CAD Integration Model Edits
Multiple Geometry Links for CAD Integration
CAD Integration Functionality
Healing with CAD Integration
Important Geometry Options for CAD Integration

Bi-Directional CAD Integration

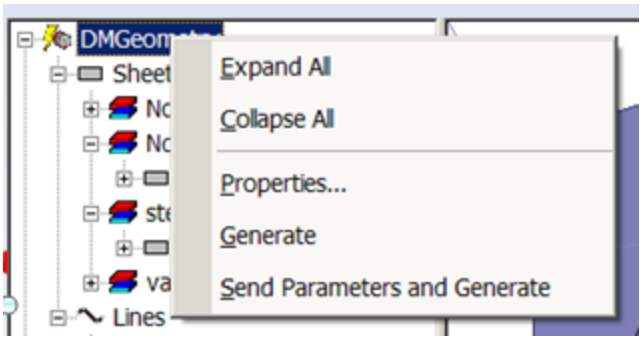
Ansys Electromagnetics CAD Integrations uses Refresh (or Generate) CAD Model to pass updates.

For example, you can make an edit in a CAD application and either run **Refresh** on an Ansys Electromagnetics Geometry Cell or run **Generate** on the UDM in the Ansys Electromagnetics window



Refresh pulls the current state of CAD model (geometry, parameters, materials etc) and updates the corresponding data in the Ansys Electromagnetics application.

If you edit UDM (CAD) parameters in the Ansys Electromagnetics modeler window you can run the **Send Parameters and Generate** command



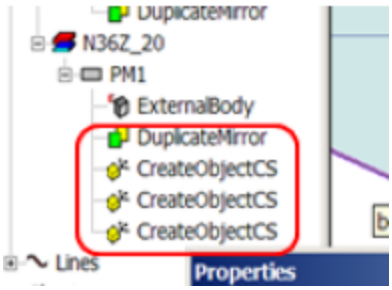
The command passes the edited parameters to the linked CAD application and then pulls corresponding CAD geometry.

Related Topics

Ansys Electromagnetics CAD Integration Through Workbench
CAD Integration and Geometry Sharing
CAD Integration Model Edits
Multiple Geometry Links for CAD Integration
CAD Integration Functionality
Healing with CAD Integration
Important Geometry Options for CAD Integration

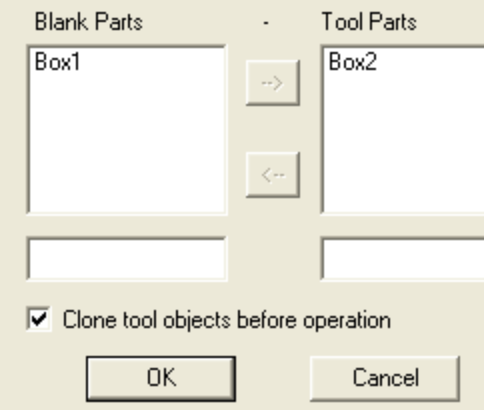
CAD Integration Model Edits

Several modeling operations are allowed on a CAD model in the Ansys Electromagnetics Modeler window. Operations will included in the History tree and retained during model **Refresh**.



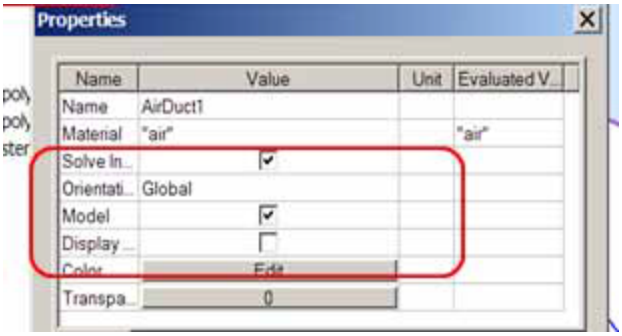
The following operations are not allowed:

- Non-history tree operations like Heal or defeature.
- Operations which use UDM parts as Tool such as Sweep, or Boolean operations like Split or Unite (but allowed when you select the clone tool option).



The following part attributes can be modified for UDM parts.

- Model/Non Model flag
- Solve Inside flag
- Part orientation
- Color
- Display Wireframe



It is not possible to delete individual parts of UDM

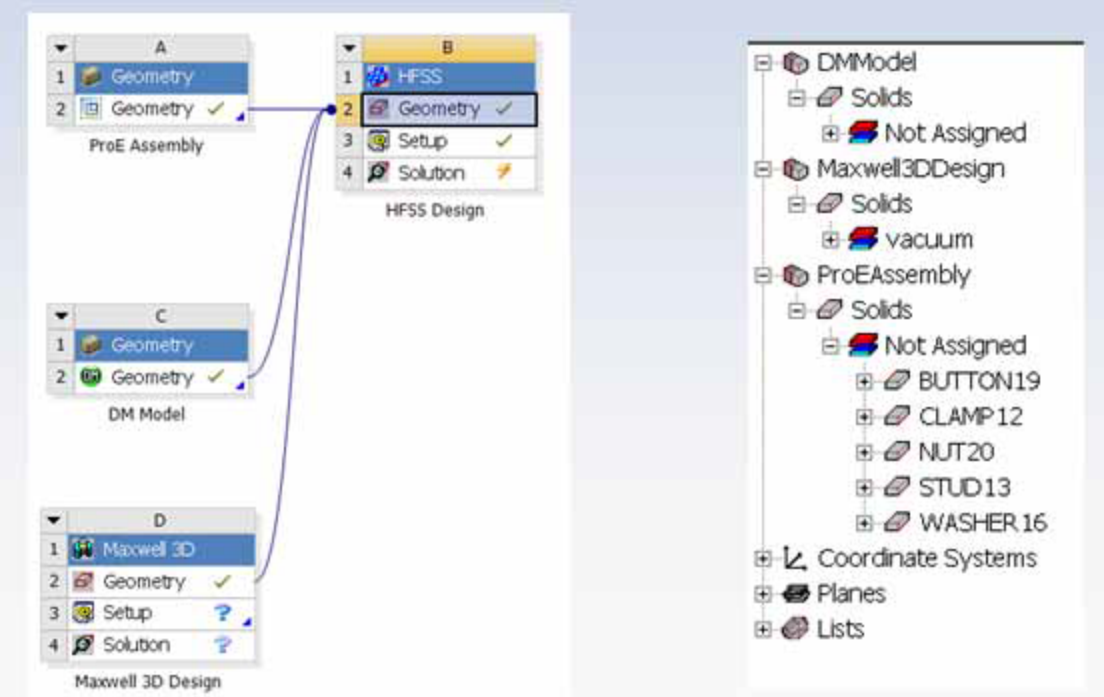
Related Topics

Ansys Electromagnetics CAD Integration Through Workbench
CAD Integration and Geometry Sharing
Bi-Directional CAD Integration
Multiple Geometry Links for CAD Integration
CAD Integration Functionality
Healing with CAD Integration
Important Geometry Options for CAD Integration

Multiple Geometry Links for CAD Integration

With CAD Integration you can utilize geometry from multiple upstream sources. The Source can be from any CAD or Ansys Electromagnetics product. This creates a user-defined model (UDM) for each geometry source.

For example, the following figure shows a DM Model, a Maxwell model, and ProE (Creo Parametric) model linked to HFSS in Workbench, and displayed in the HFSS History tree as three UDMs.



Related Topics

Ansys Electromagnetics CAD Integration Through Workbench
CAD Integration and Geometry Sharing
Bi-Directional CAD Integration
CAD Integration Model Edits
CAD Integration Functionality
Healing with CAD Integration
Important Geometry Options for CAD Integration

CAD Integration Functionality

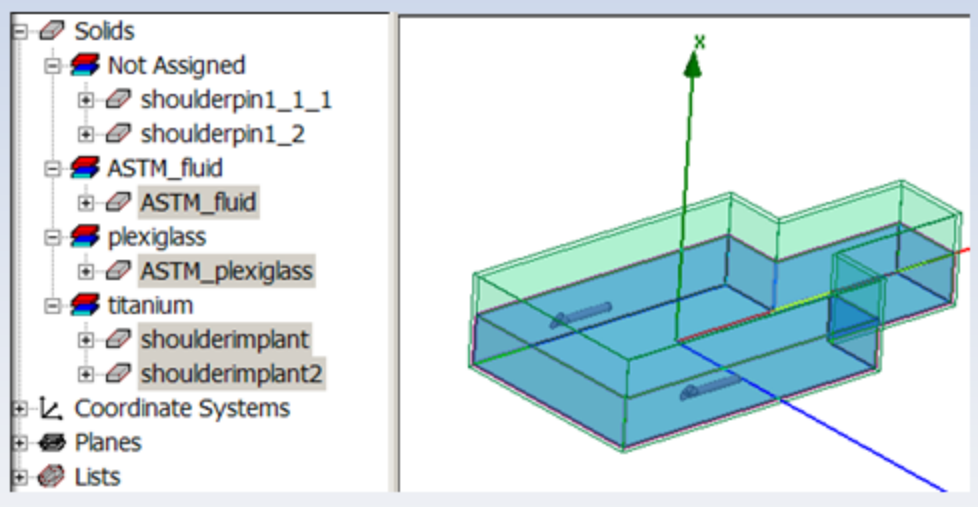
- WB Update Project
- WB Update All Design Points

- DX analysis
- Parametric Analysis with DSO
- Animation
- Geometry
- Field Plots

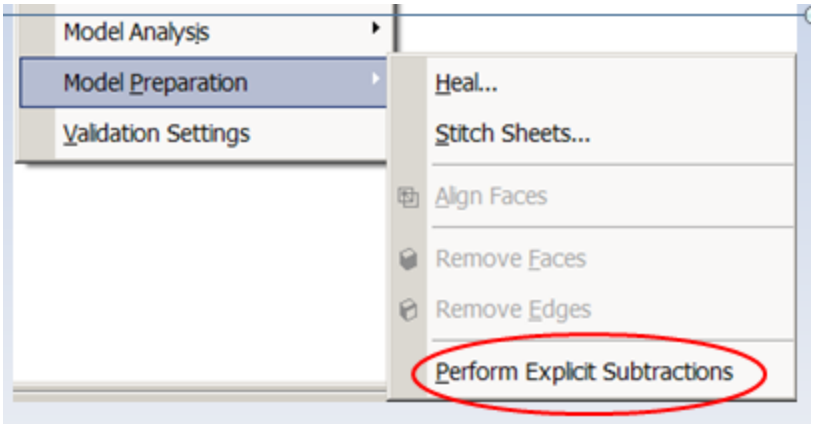
For Explicit Subtraction

- Ansys Mechanical does not do implicit subtraction
- Ansys Electromagnetics modeler command **Modeler>Model Preparation>Perform Explicit Subtraction** can do explicit subtractions before sending geometry to ANSYS.

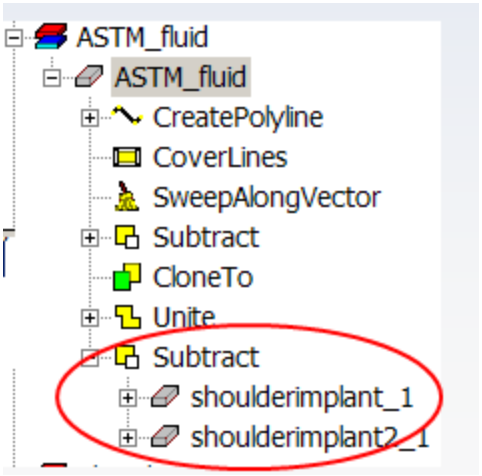
For example, consider the following model.



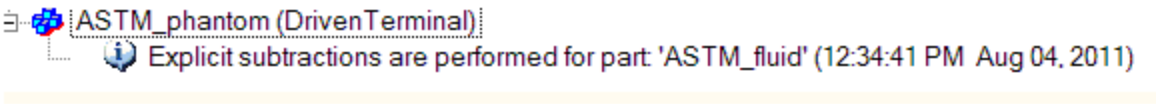
Perform Explicit Subtractions can be performed.



The results appear in the History tree as shown:



The Message window reports this action.

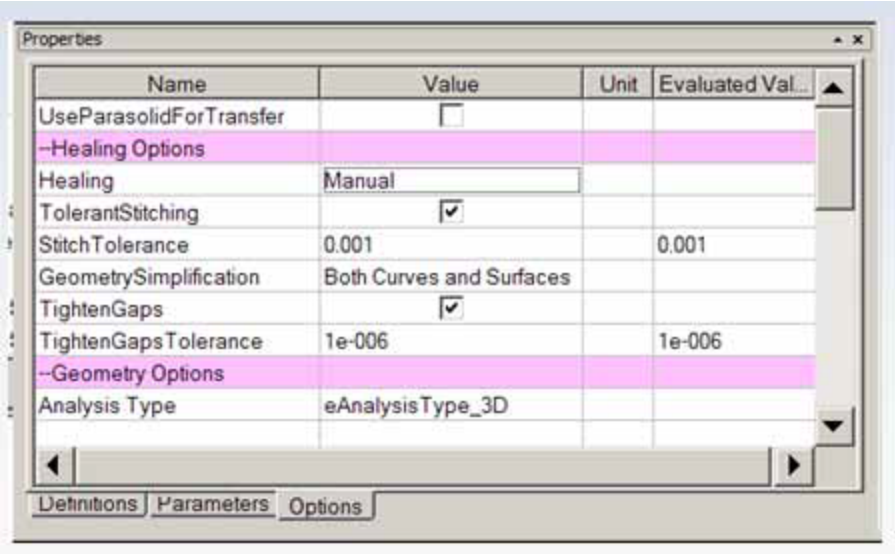


Related Topics

User Defined Model (UDM) for Ansys WB Integration
UDM compared to User Defined Primitives
Insert UDM Command on Draw Menu
UDM Properties
Library of Models for CAD Integration
Ansys to Ansys Electromagnetics Geometry Transfer
CAD Integration Material Assignment Transfer
Geometry Transfer through Ansys DesignModeler (DM)

Healing with CAD Integration

It is not possible to use the Heal command in Ansys Electromagnetics Modeler. Instead similar healing options are available under UDM properties option tab.



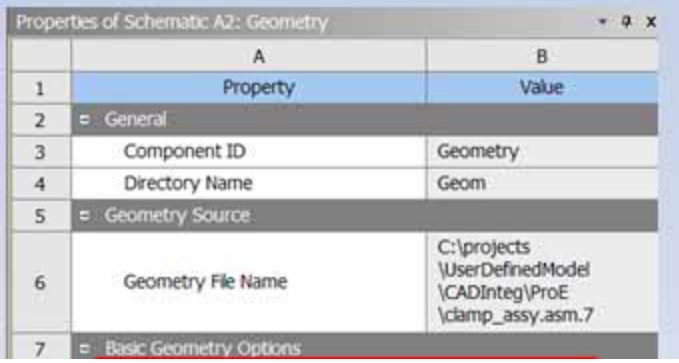
The Healing options are: None, Auto and Manual. By default healing is off (None) and should be turned on only if required.

Related Topics

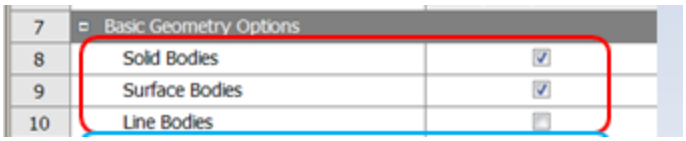
Ansys Electromagnetics CAD Integration Through Workbench
CAD Integration and Geometry Sharing
Bi-Directional CAD Integration
CAD Integration Model Edits
Multiple Geometry Links for CAD Integration
CAD Integration Functionality
Important Geometry Options for CAD Integration

Important Geometry Options for CAD Integration

Select a Geometry Cell in Workbench to see options in Properties window.

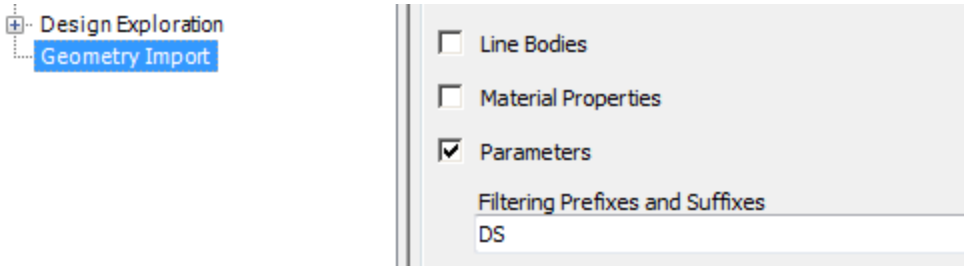


- Control dimension of bodies coming from CAD



- Make sure parameters is checked and parameter key (filter) is appropriate to bring CAD parameters.

This (default) parameter key filtering means that only parameters whose names start with DS will come through. Notice that the Parameter Key value (default DS) is set via the Workbench **Tools>Options...** dialog box, under Parameters, in the Filtering Prefixes and Suffixes field.



- Attributes key should be empty or Color to bring in CAD Colors

- Material properties must be checked to bring in the material assignment.

- The Mixed import resolution option is used to resolve parts with mixed dimension (typically from Pro/E)

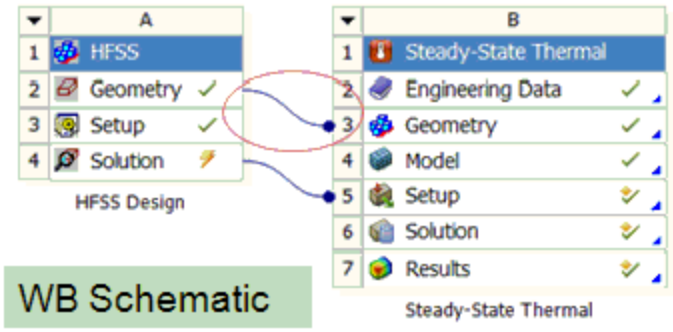
Refer to Ansys Workbench Help for details.

Related Topics

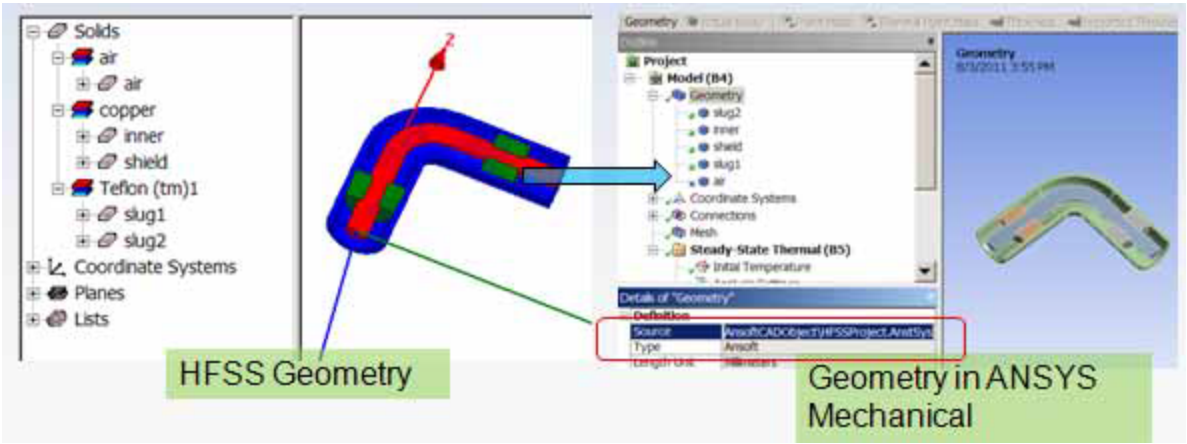
Ansys Electromagnetics CAD Integration Through Workbench
CAD Integration and Geometry Sharing
Bi-Directional CAD Integration
CAD Integration Model Edits
Multiple Geometry Links for CAD Integration
CAD Integration Functionality
Healing with CAD Integration

Ansys Electromagnetics to Ansys Geometry Transfer

Ansys Electromagnetics CAD Integration transfers model information based on Workbench links.

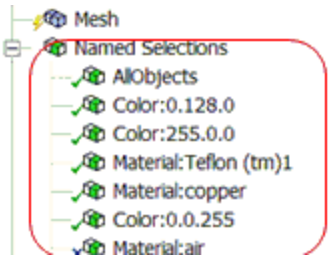


The following figure shows how the information is transferred between simulators.

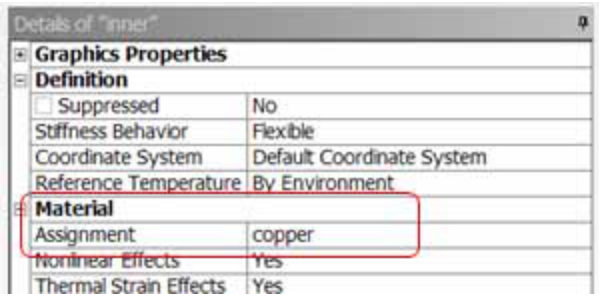


The information transferred includes:

- Geometry
- Ansys Electromagnetics lists and material assignment as Named Selection



- Material assignment



The CAD Integration geometry link is

- Dynamic because you can get updated geometry from Ansys Electromagnetics
- Associative because IDs persist between Ansys Electromagnetics model and Ansys model during model refresh.

  Boundary conditions in Ansys are preserved.

Related Topics

User Defined Model (UDM) for Ansys WB Integration

UDM compared to User Defined Primitives

Insert UDM Command on Draw Menu

UDM Properties

Library of Models for CAD Integration

CAD Integration Material Assignment Transfer

Geometry Transfer through Ansys DesignModeler (DM)

CAD Integration Functionality

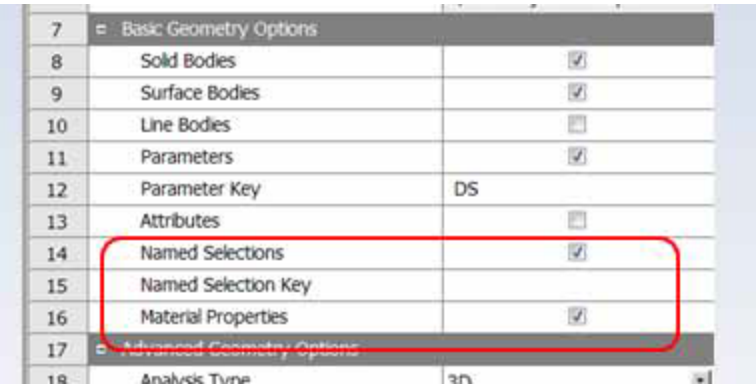
CAD Integration Material Assignment Transfer

Prerequisite for assignment transfer

- Engineering Data should have materials used in Ansys Electromagnetics model and material names should match with case.
- Material Properties in Geometry Options must be checked

Prerequisite for transfer as named selection

- Named Selection in Geometry Options must be checked



- Named Selection Key should either be empty or include ‘Material’

Related Topics

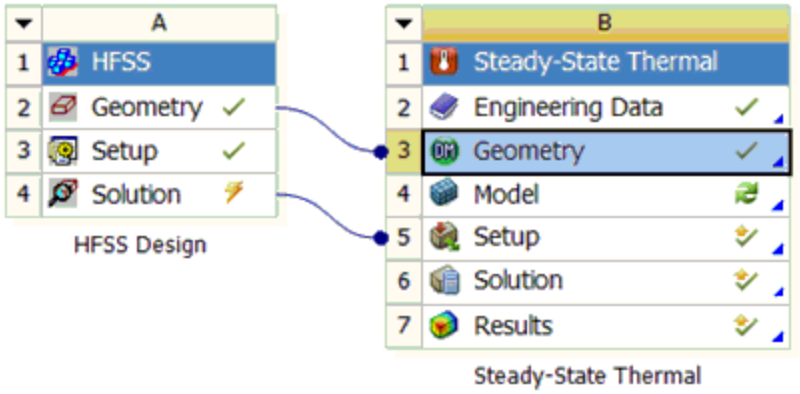
User Defined Model (UDM) for Ansys WB Integration
UDM compared to User Defined Primitives
Insert UDM Command on Draw Menu
UDM Properties
Library of Models for CAD Integration
Ansys to Ansys Electromagnetics Geometry Transfer
Geometry Transfer through Ansys DesignModeler (DM)
CAD Integration Functionality

Geometry Transfer through Ansys DesignModeler (DM)

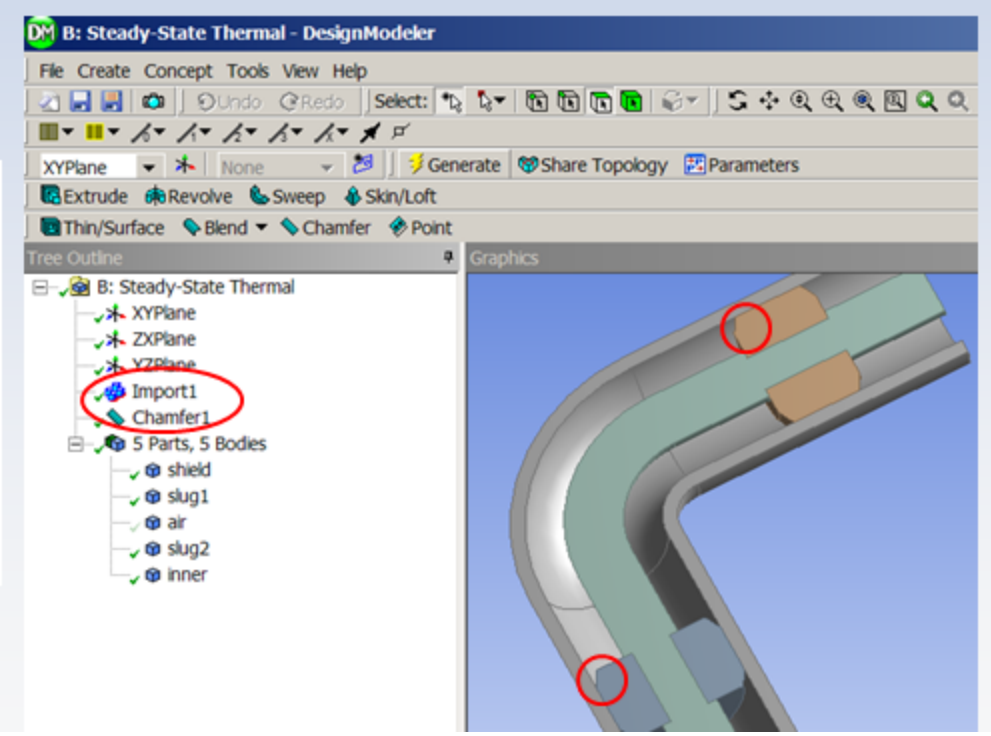
It is possible to edit geometry in Ansys DesignModeler (DM) before consuming in Mechanical

- Useful when geometry needs preprocessing for ANSYS simulation
- Can take advantage of geometry commands available in DM

For example, consider an HFSS model linked to DM through the Workbench.



In this case, the following figure shows how a chamfer operation on geometries is imported.



It is possible to edit geometry in Ansys DesignModeler (DM) before consuming in Mechanical

- Useful when geometry needs preprocessing for ANSYS simulation
- Can take advantage of geometry commands available in DM

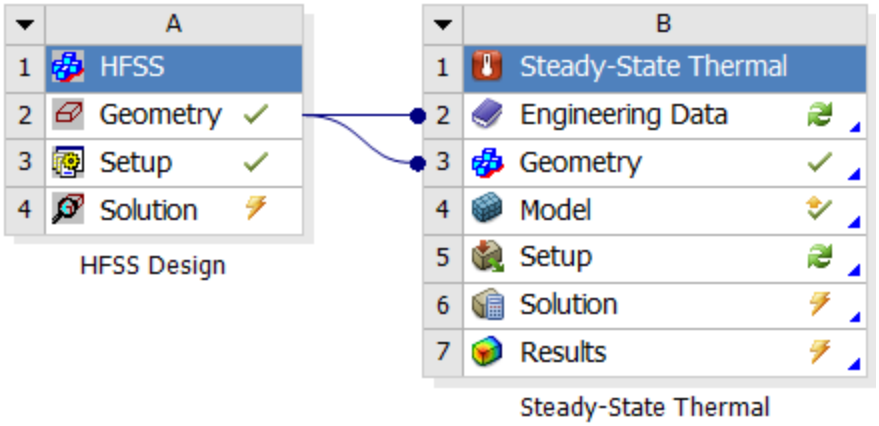
Related Topics

User Defined Model (UDM) for Ansys WB Integration

UDM compared to User Defined Primitives

Workbench Material Data Transfer

A link can now be created from an Ansys Electromagnetics Desktop project (HFSS, Maxwell, etc.) geometry cell to a downstream Engineering Data (ED) cell. When the downstream ED cell is refreshed, it will be updated with materials data (names and properties defined in the upstream project) that are used by the upstream project. This ensures that the geometry-to-geometry link connecting the same two systems, which specifies the material assignments for various parts, refers to the correct materials and their properties.



Detailed Behavior

The geometry-to-ED link can be created, for example, by dragging the Ansys Electromagnetics Desktop project geometry cell to the ED cell of a downstream system. When the downstream system's ED cell is refreshed, the upstream system generates a MatML formatted XML file which is used by Workbench to update the downstream system. This file is stored in the workbench project files according to design point. Note that Workbench ED cell material names are case-sensitive, as are material names within the Ansys Electromagnetics Desktop designs.

For the material assignments to work with the ED in the downstream model, the downstream geometry cell must have the **Material Properties** checkbox selected/checked. A convenient way to do this is to have this property checked by default, by setting it in **Tools>Options>Geometry Import**. Designs created after this default option is set will have Material Properties checked automatically for the geometry cell.

When the downstream model cell is refreshed or updated, the material properties from the ED in the same system are used in the model's materials, based on the material assignments set during the geometry cell refresh. Note that if material assignment fails, the Workbench default material Structural Steel will be used.

Workbench resolves name conflicts by changing the name of any materials being transferred over from the upstream system. For example, if the upstream Ansys Electromagnetics Desktop design uses a material named “Structural Steel”, this will conflict with the existing downstream default material with the same name. The update will result in the Ansys Electromagnetics Desktop material being named “Structural Steel 2", which will silently break the material assignment, because the resulting model will refer to "Structural Steel" as it originally did, and use the default workbench property values, instead of using the properties which were supplied by Ansys Electromagnetics Desktop. If this occurs, rename the material in the Ansys Electromagnetics Desktop project (in such a way that there is no more conflict), reset the downstream ED cell, reset the downstream geometry cell, and then refresh those cells in the downstream system.

Note that some materials may not satisfy all of the requirements for the downstream system physics. For example, vacuum has no thermal conduction coefficient, but downstream steady state thermal systems require a non-zero value for this property. Material properties in the Workbench ED cell of the downstream system can be edited/modified after the refresh to ensure that validity criteria can be met. Any property added to a material in the downstream ED will be preserved in subsequent ED cell updates. However, any existing property (i.e., a property that was given by the upstream system material definition) that is edited in the downstream ED will be overwritten by a subsequent ED cell update.

Related Topics

Ansys Electromagnetics CAD Integration Through Workbench

Bi-Directional CAD Integration

CAD Integration Model Edits

Multiple Geometry Links for CAD Integration

CAD Integration Functionality

Healing with CAD Integration

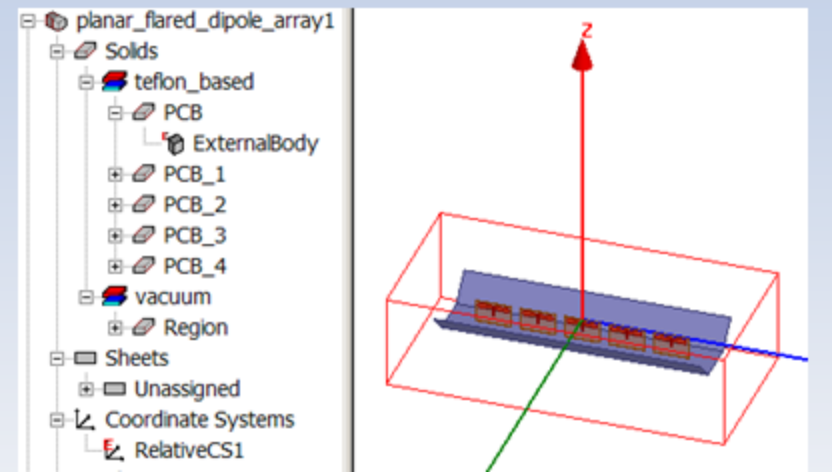
Important Geometry Options for CAD Integration

User Defined Model (UDM)

A User Defined Model (UDM) is collection of externally defined parts imported into an Ansys EM 3D Modeler or created using a C or Python script.

- UDM includes part attributes (like name, color etc) and material assignment
- UDM can also have external coordinate systems and corresponding planes
- UDM parts can be parameterized and manipulated in Ansys Electromagnetics modeler just

like any other part



UDM can either represent Static geometry models or

- Dynamic links to models of external geometry editors
- Used for supporting CAD integration in WorkBench

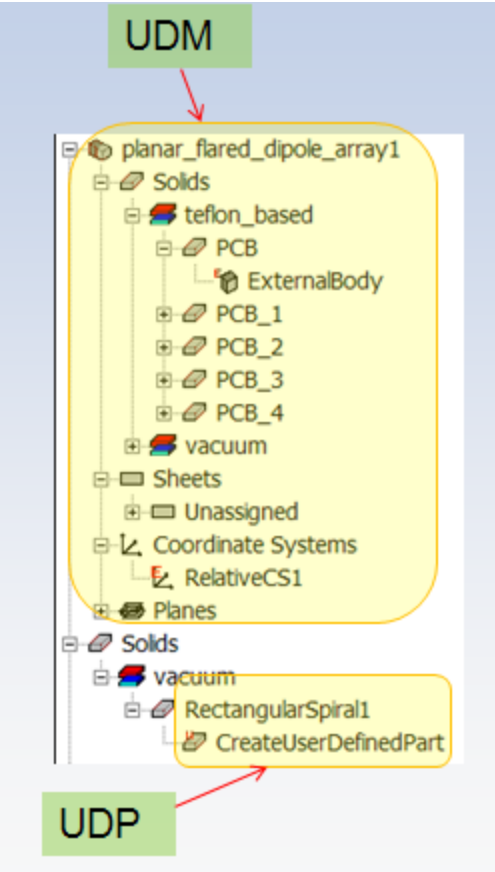
  UDM uses same plugin technology as User Defined Part (UDP)

Related Topics

UDM compared to User Defined Primitives
Insert UDM Command on Draw Menu
UDM Properties
Library of Models for CAD Integration
Ansys to Ansys Electromagnetics Geometry Transfer
CAD Integration Material Assignment Transfer
Geometry Transfer through Ansys DesignModeler (DM)
CAD Integration Functionality

UDM compared to User Defined Primitives

User Defined Models (UDM) resemble User Defined Primitives (UDP):



- Ansys Electromagnetics products can be extended by users through new UDMs.
- UDM plugins are discovered by searching standard directory paths.
- Plugins for static UDM can build model using 'callback interfaces' (like create-box, create-cylinder, subtract etc) similar to UDP.
- UDMs run inside Ansys Electromagnetics application.
- UDMs provide geometry, topology, persistence and parameters.

In contrast to UDP:

- UDM provides multiple Parts/CS/etc.

  UDP provides primitive operation only for a single Part.
- UDM provides part attributes and material assignment.

  UDP does not define part attributes or material.

UDM Properties have four tabs – Definitions, Parameters, Options and Info.

- Definition tab has:

  UDM name

  Coordinate system used to position UDM

  May have external reference to file
- Info tab has:

UDM dll name, dll location, version etc.

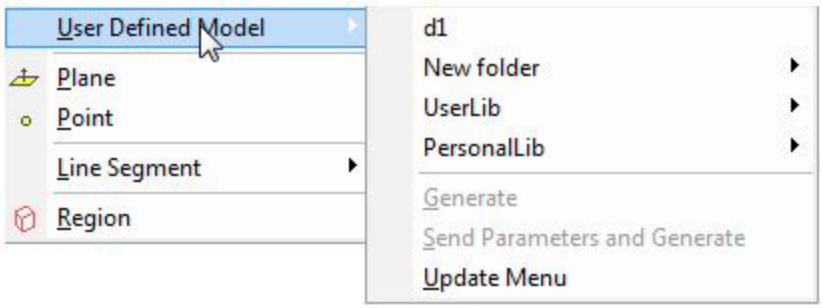
- Option tab:
  may have options

Related Topics

User Defined Model (UDM) for Ansys WB Integration
Insert UDM Command on Draw Menu
UDM Properties
Library of Models for CAD Integration
Ansys to Ansys Electromagnetics Geometry Transfer
CAD Integration Material Assignment Transfer
Geometry Transfer through Ansys DesignModeler (DM)
CAD Integration Functionality

Insert UDM Command on Draw Menu

To insert a UDM into a design, use the User Defined Model command on the Draw menu for the Modeler window.

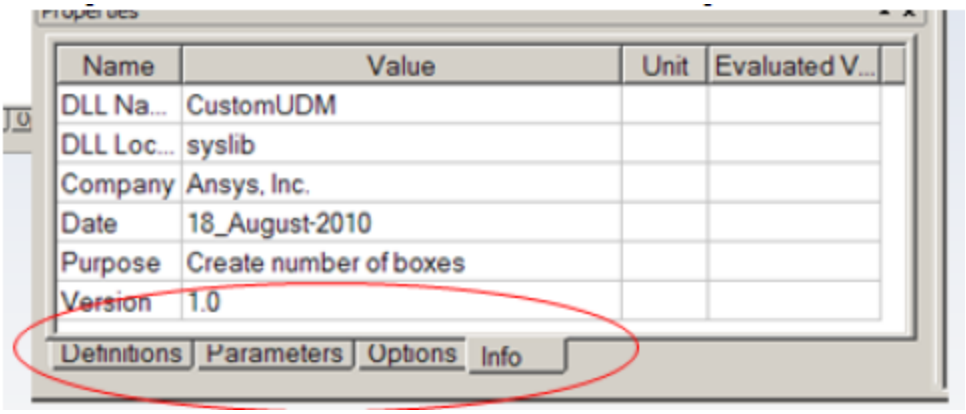


Related Topics

User Defined Model (UDM) for Ansys WB Integration
UDM compared to User Defined Primitives
UDM Properties
Library of Models for CAD Integration
Ansys to Ansys Electromagnetics Geometry Transfer
CAD Integration Material Assignment Transfer
Geometry Transfer through Ansys DesignModeler (DM)
CAD Integration Functionality

UDM Properties

UDM Properties have four tabs – Definitions, Parameters, Options and Info.



Definition tab has

- UDM name
- Coordinate system used to position UDM
- May have external reference to file

Info tab has:

- UDM dll name, dll location, version etc

Option tab:

- may have options if any

UDM Parameters

UDM Part Edits

Related Topics

User Defined Model (UDM) for Ansys WB Integration

UDM compared to User Defined Primitives

Insert UDM Command on Draw Menu

Library of Models for CAD Integration

Ansys to Ansys Electromagnetics Geometry Transfer

CAD Integration Material Assignment Transfer

Geometry Transfer through Ansys DesignModeler (DM)

CAD Integration Functionality

UDM Parameters

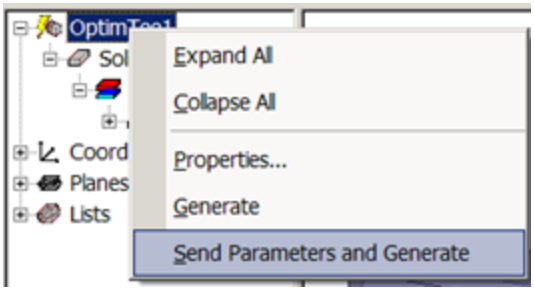
UDM geometry can be manipulated through its parameters

- Can be mapped to design or project variable for animation, parametric analysis
- IDs are persisted (allowing to retain boundaries) during parameter edit

UDM geometry is not dynamically updated upon parameter edits

- UDM shows a lightning bolt icon by the model name when parameters are edited.

- You must run the **Send Parameters and Generate** command to synchronize parameters with geometry



Related Topics

UDM Properties

UDM Part Edits

UDM Part Edits

Several modeling operations are allowed on UDM parts

- Operations will be part of history tree and retained during model refresh

The following operations are not allowed

- Non history tree operations like healing, defeature.
- Operations which use UDM parts as tool, such as sweep or boolean (but allowed when clone tool option is selected)

Following part attributes can be modified for UDM parts

- Model/Non Model flag
- Solve Inside flag
- Part orientation

Related Topics

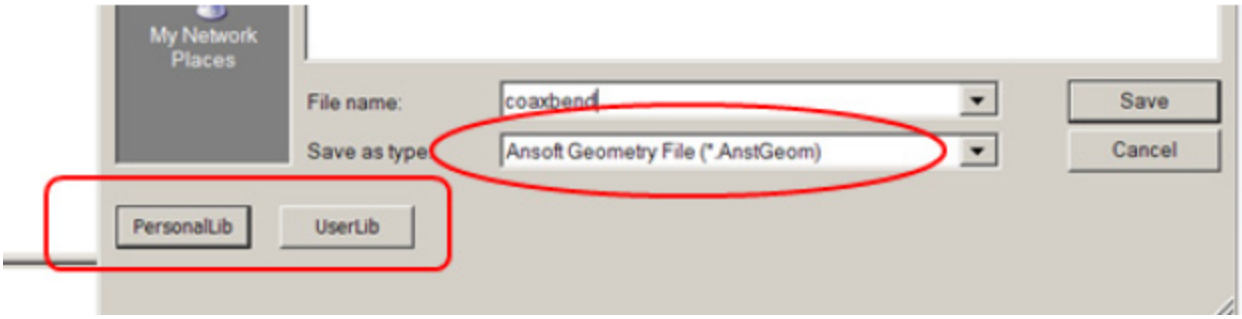
UDM Properties

UDM Parameters

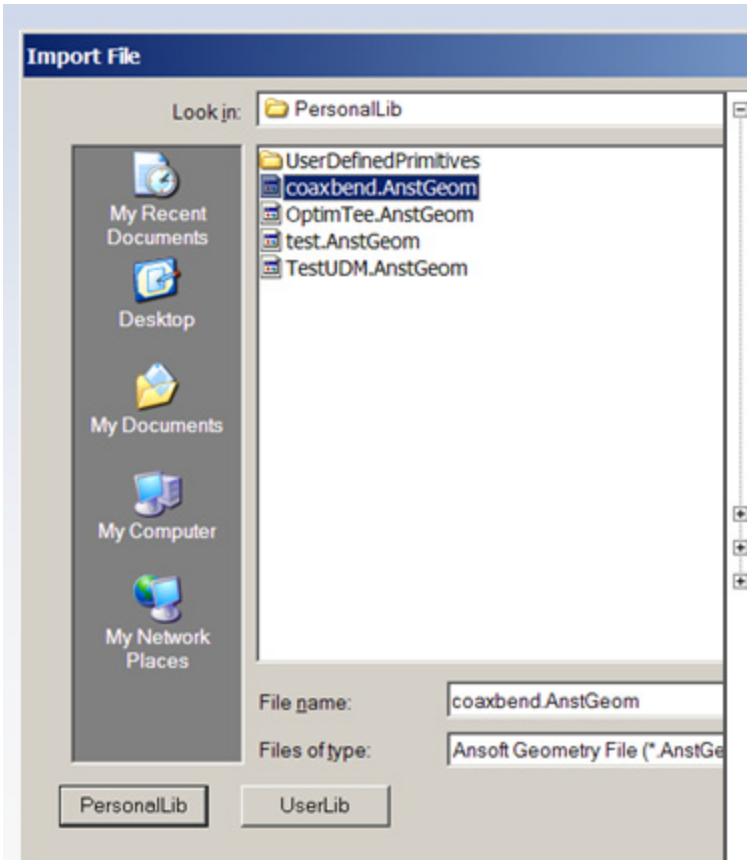
Library of Models for CAD Integration

UDM technology allows library of models

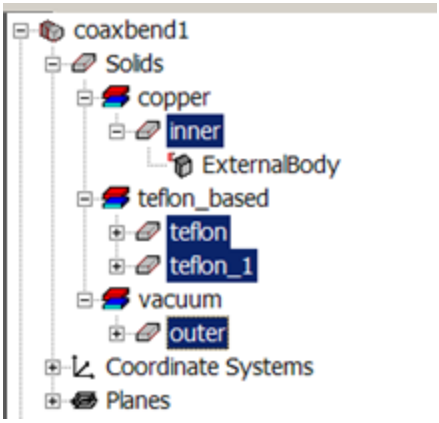
- Any Ansys Electromagnetics model can be exported as ‘Ansoft Geometry File’



- An Ansoft Geometry File can be imported back as a UDM.



- Any Design/project variables associated with model are brought in as UDM parameters.



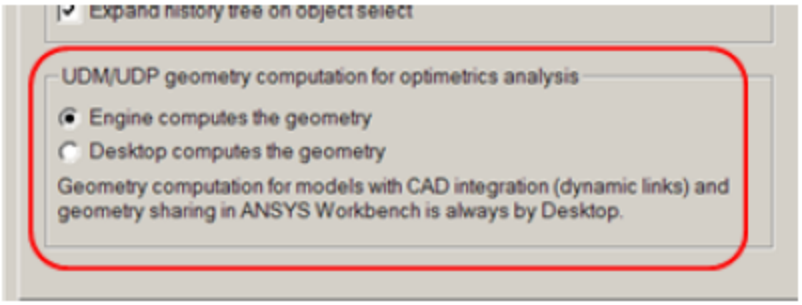
Geometry computation for UDM (and also UDP) can be specified in the Modeler options as either done on.

- Engine side (default):

  Requires deployment of UDM on each node
- Desktop side:

  UDM need not be deployed on each engine

  Desktop will be busy during parametric analysis



Related Topics

User Defined Model (UDM) for Ansys WB Integration

UDM compared to User Defined Primitives

Insert UDM Command on Draw Menu

UDM Properties

Ansys to Ansys Electromagnetics Geometry Transfer

CAD Integration Material Assignment Transfer

Geometry Transfer through Ansys DesignModeler (DM)

CAD Integration Functionality

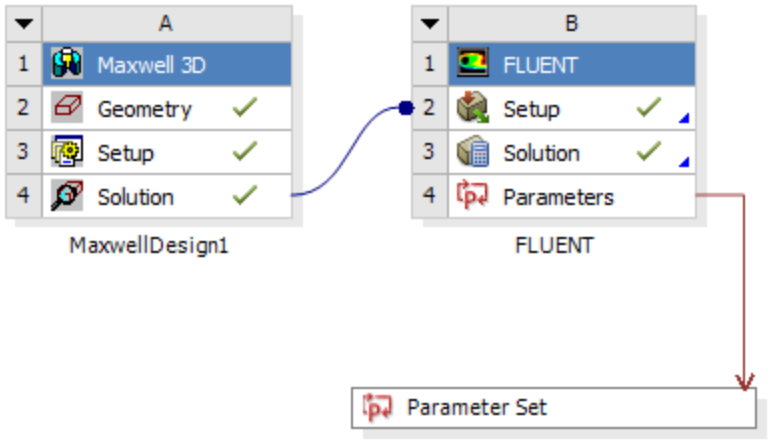
Feedback Iterator

Background Information

Prior to the introduction of the Feedback Iterator, ANSYS Workbench supported a two-way loose-coupling protocol with Ansys Electromagnetics products.

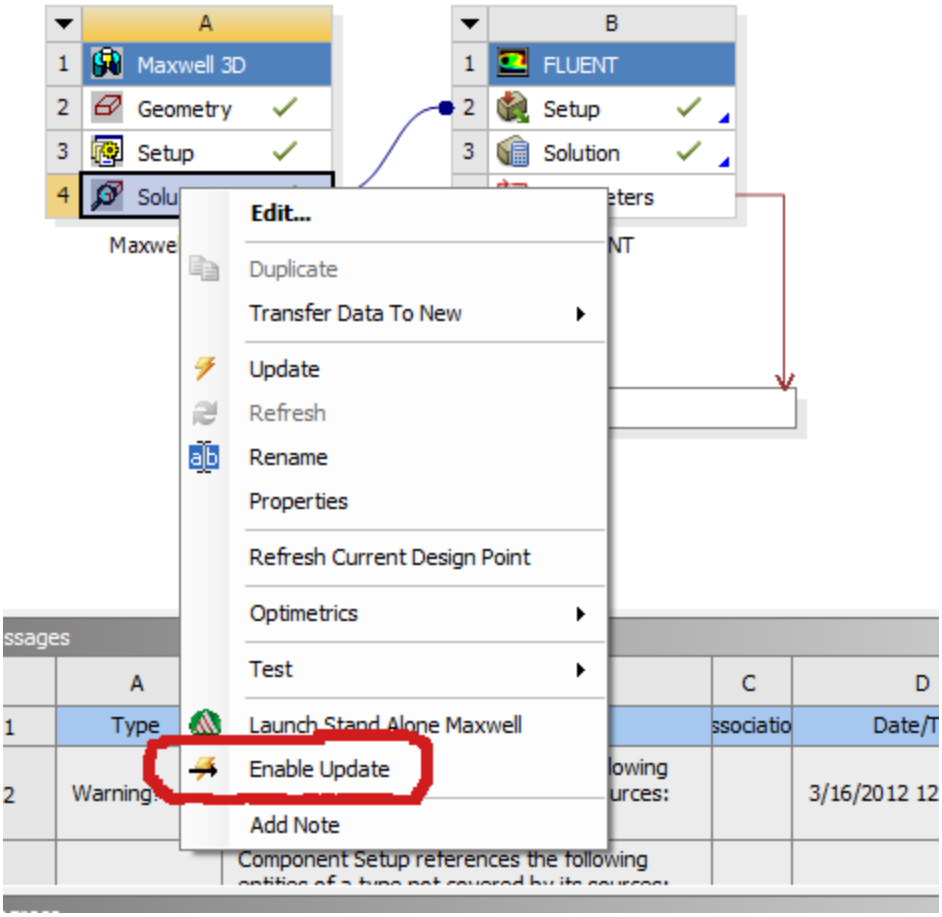
- System Coupling uses the word “coupling” to mean low-level solver coupling. Existing Ansys Electromagnetics coupling is “loose” when compared to System Coupling and is limited to file transfers at the end of a complete solve in a stand-alone system/product. No communication occurs during a solve.
- The coupling is two-way:

  The one way portion (Upstream source component to Downstream target component) is handled via existing Workbench data/transfer connection mechanism.



The round trip is handled by a separate protocol agreed upon by the participating systems when the downstream system exports a set of file to a location specified by the upstream system (via its one way transfer). This exported data is then incorporated by the upstream system in its next update.

If the user chooses to run the next coupled solve iteration, the user invokes the Enable Update command as shown below and then updates all systems involved. These steps (Enable Update, Update Project) are continued as long as needed



The **Feedback Iterator** system automates this process as detailed in the following related topics:

The Feedback Iterator System

Feedback Iterator in Use

Feedback Iterator Component Properties

Feedback Iterator GUI Operations

Resetting the Feedback Iterator

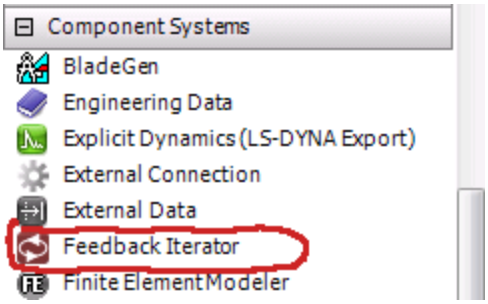
Callback Interface

Example Scenarios for Feedback Iterator

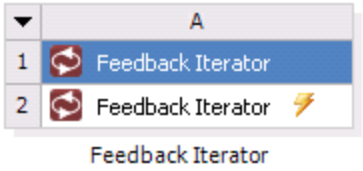
The Feedback Iterator System

Ansys Electromagnetics Suite provides the Feedback Iterator system for automating the manual steps needed for driving a feedback utilizing system-pair to convergence. In addition to automating feedback incorporation over a user-specified number of iterations, the Feedback Iterator also allows you to control the number of iterations (decide iteration termination criteria) and set target temperature and displacement convergence criteria.

The Feedback Iterator appears in the Workbench Toolbox user interface under Component Systems.

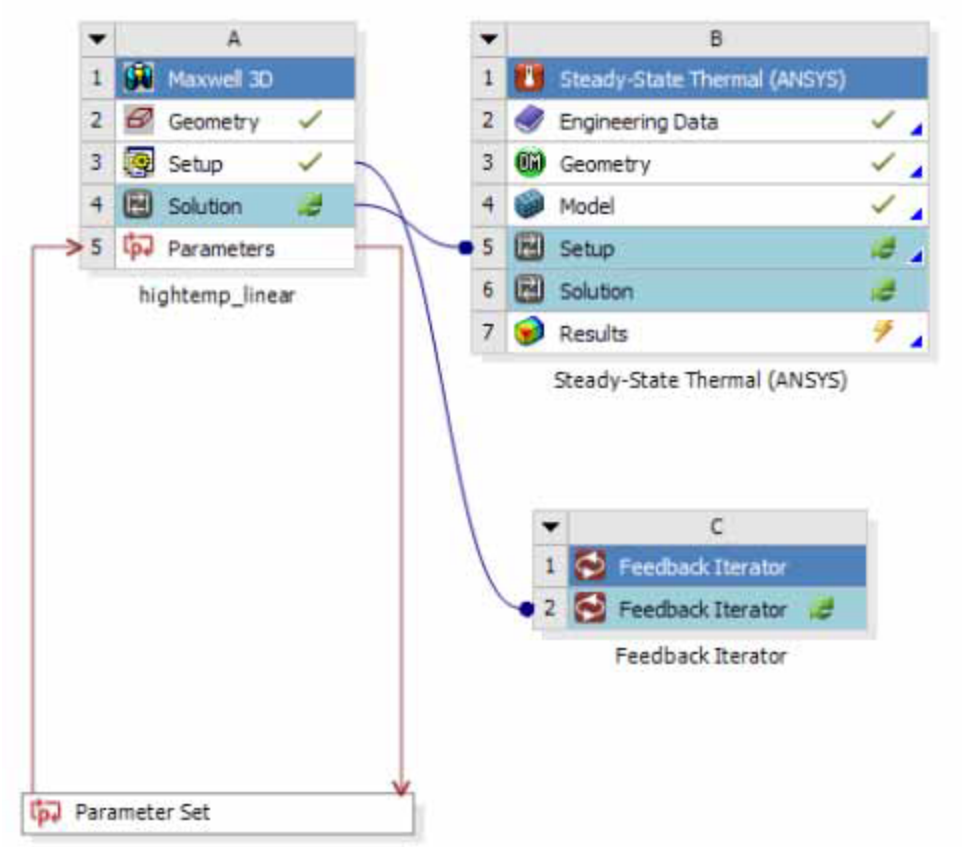


If you drag and drop an instance of the Feedback Iterator into Workbench Schematic, the default system name is Feedback Iterator.



Feedback Iterator in Use

You attach the Feedback Iterator system to the Workbench schematic like any other system. For example, given a Maxwell system providing AnsoftHeatLossData to a Thermal system, if you drop the Maxwell setup cell onto the Feedback Iterator component, the following links result



While the examples deal with Maxwell and Steady-State Thermal. the Feedback Iterator add-in functions the same way with Maxwell or HFSS as sources and Fluent or Structural as targets – essentially all systems that participate in Electromagnetics two-way loose coupling.

Feedback Iterator Component Properties

The Feedback Iterator's primary role is to control iterative solves and its properties support this end.

For Maxwell and HFSS, the Feedback Iterator properties are as shown below:

	A	B
1	Property	Value
2	⊟ General	
3	Component ID	FeedbackIterator
4	Directory Name	FeedbackIteratorComponent
5	⊞ Notes	
7	⊞ Used Licenses	
9	⊟ Iterations	
10	Iterations Completed	0
11	Max Iterations	100
12	⊞ Callback	
14	⊟ Temperature Convergence	
15	Target Delta Temperature %	5
16	Latest Delta Temperature %	Not Available
17	⊟ Displacement Convergence	
18	Target Delta Displacement %	5
19	Latest Delta Displacement %	Not Available

- **Iterations Completed**: displays the number of iterations completed. A “read-only” property.
- **Max Iterations**: sets the maximum number of iterative solve loops to perform before terminating iterations in case the **Target Delta Temperature %** or **Target Delta Displacement %** is not achieved. Default value is 100.
- **Target Delta Temperature %**: specifies the maximum **Target Delta Temperature %** which signifies convergence. Default value is 5%. Value must be 0.01 or greater.

Note	• The delta temperature error is calculated base on the SI unit: Kelvin. • The Maxwell and HFSS design Profile tabs display absolute/relative delta data, while the delta temperate in Workbench means the relative delta (expressed as %).

- **Target Delta Displacement %**: specifies the maximum **Target Delta Displacement %** which signifies convergence. Default value is 5%. Value must be 0.01 or greater.

The Feedback Iterator properties also display the most recently achieved **Latest Delta Temperature %** and **Latest Delta Displacement %** values, allowing the user to manually abort when suitably satisfied with the achieved deltas.

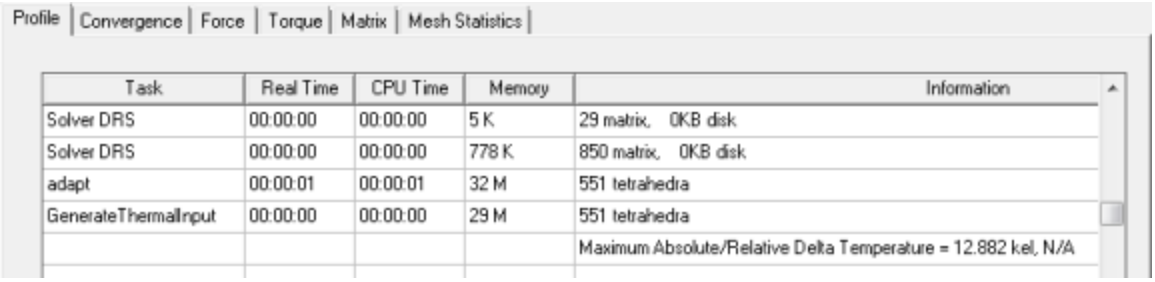
Note	Both the Temperature and Displacement Convergence properties sections are always shown irrespective of which feedback types are actually enabled for the project.

When we run a two-way feedback simulation from WB, and keep the Maxwell or HFSS Profile tab open (right-click on Results in the Project Manager and select Solution Data, then select Profile tab), you will see that the 3D solver keeps track of two feedback related quantities:

- Maximum Absolute/Relative Delta Temperature (if temperature feedback is enabled)

Note	The delta temperature error is calculated base on the SI unit: Kelvin.

- Maximum Absolute/Relative Delta Displacement (if displacement feedback is enabled)

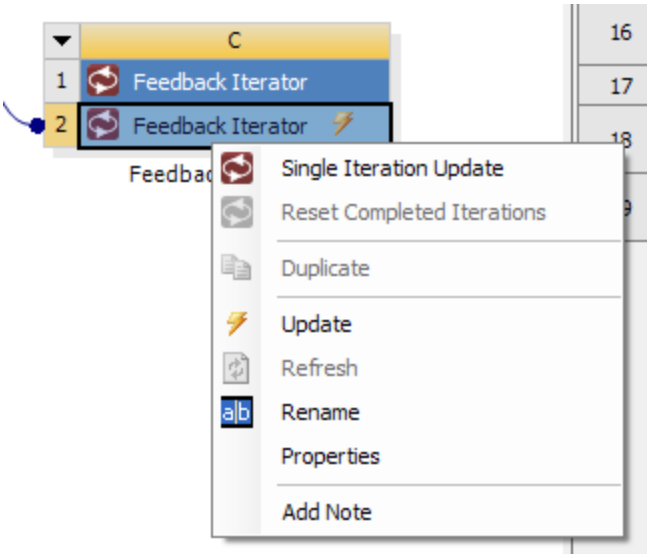


These signify the maximum difference on the solution mesh of the quantity in question from one feedback iteration to the next (thus the value is only available from iteration #2 onward). This data is the basis for the convergence controls in the Feedback Iterator. These controls allow the user to base the termination of the feedback iterations based on a target max delta T and/or max delta displacement. Thus the target max delta forms the convergence condition.

Feedback Iterator GUI Operations

When you project is ready to run, right-clicking on the Feedback Iterator component and selecting **Single Iteration Update** allows you to run a single iteration worth of updates. A Single Iteration Update operates as follows:

- The Electromagnetics product system (Maxwell, etc.) is updated, which will incorporate any previous feedback from the downstream system.
- The downstream system (Fluent, Thermal, etc.) is updated.
- All coupled systems: Electromagnetics product, Electromagnetics product Downstream and the Feedback Iterator will then be in the UpToDate state.
- The Feedback Iterator's **Iterations Completed** property will increment by 1.



Right-clicking on the Feedback Iterator component and selecting **Update** allows you to run iterations automatically. Iterations continue to run either until convergence criteria are met, or until the **Max Iterations** value is reached.

Note	After getting temperature feedback from Fluent, Maxwell solving starts from the *last adaptive*, which means there is NO adaptive pass solving anymore. Consequently, instead of looking for the expression cache value in the adaptive solution, users should look for it in the *last adaptive* solution. For *Last Adaptive*, the expression cache value can be displayed from **Results > Create Fields Report > Data Table**.

Resetting the Feedback Iterator

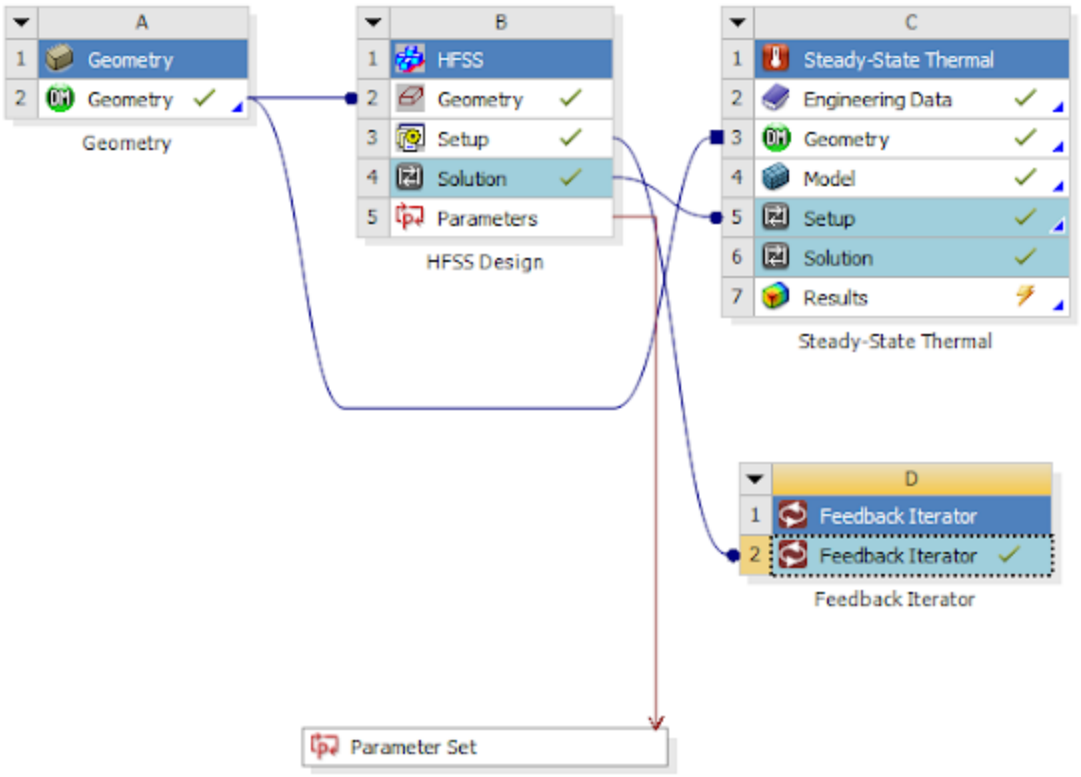
Resetting the iterator essentially means starting from iteration-0 conditions, that is:

- No Displacement or Temperature feedback from another system is incorporated.
- The first iteration solution will be purely an Ansys EM analysis solution.

Resetting the Feedback Iterator cannot easily be done entirely within Workbench. However, the following two processes can be used:

- Resetting via the Ansys EM user interface
- Resetting via a WB journal script

The following example uses an HFSS/Thermal project. (This same approach applies to the other Ansys EM design types.)

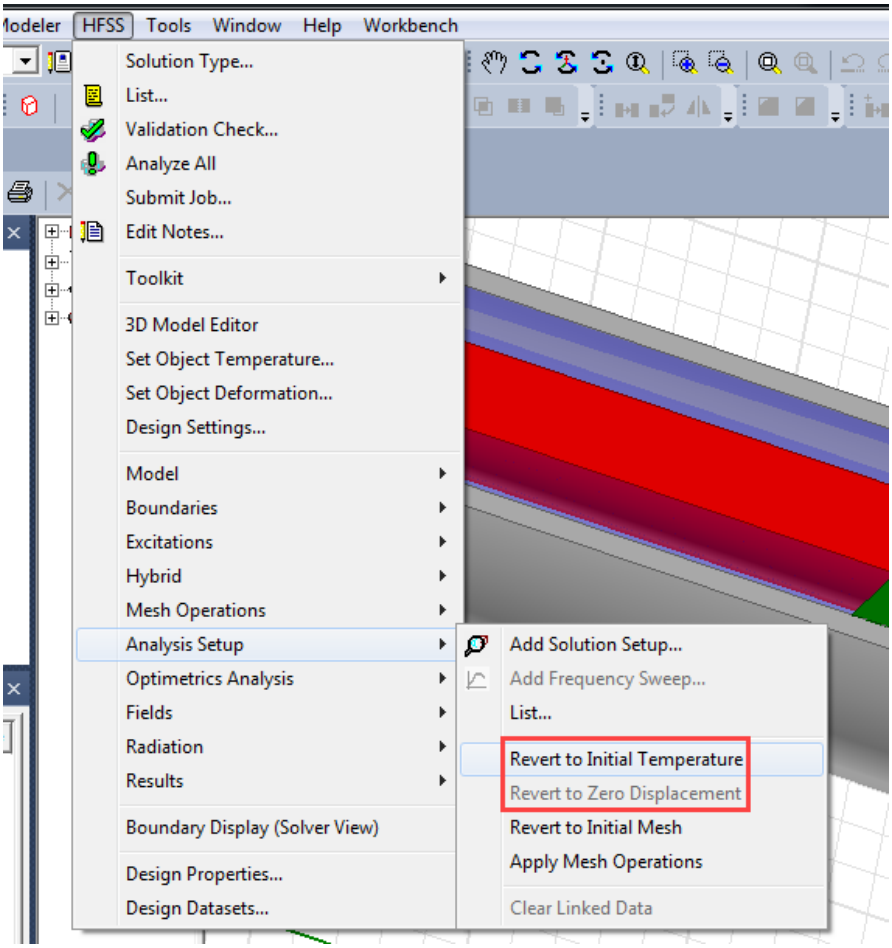


Resetting via the Ansys EM User Interface

Depending on the type(s) of feedback involved, this typically means reverting back to initial conditions for one or both of the following settings:

- **Revert to Initial Temperature**
- **Revert to Zero Displacement**

You can find these under the respective design type menu, such as HFSS shown below.



Resetting via a WB Journal Script

Note	The above Ansys EM user interface actions can be recorded into a WB journal script and played back for the easiest approach.

You also can build a script from scratch using the following as the template – simply update the system name as appropriate, include the appropriate "Revert..." commands, and either save as a script or enter into the WB Command window.

```
oDesign = Ansoft.GetDesign(GetSystem(Name="HFSS"))
oDesign.GetModule("AnalysisSetup").RevertAllToInitialTemperature()
oDesign.GetModule("AnalysisSetup").RevertAllToZeroDisplacement()
```

Callback Interface

The callback interface allows you to react to each step of the iteration process as implemented by the Feedback Iterator component. This is useful for you to implement special iterations or even solve transient simulations (with a limited scope). To this end, there are four main API features provided:

- Callback functions which the Feedback Iterator calls at various points in each iteration
- Utility functions that the Callback functions can use to extract properties of the containers being processed
- Output functions that allow you to debug the script or supply additional messages
- Limited state management allowing the script to store and retrieve state across callback functions and iterations

Related Topics

Callback and State API

Callback and State API

State is managed somewhat simply but in a limited fashion. All the API methods take a final dictionary argument. This dictionary is limited to using **string keys** and **number or string values**. Within this limitation, simply add new keys, read old keys, clear the dictionary, etc., and it will be persisted across functions calls and even across Updates. When the **Iterations Completed** property is **rest**, the dictionary is also cleared out.

It is advisable for the callback script to initialize the dictionary at Iteration 1.

All the API methods use a subset of the following arguments:

iterationNumber	An integer representing the current iteration. This always starts from 1.
ContainerList	A python list of containers (DataContainerReference). This is the entire list of the coupled containers managed by the Feedback Iterator. You typically loop through them and using the utility methods listed below, identities them.
Container	A single container that is being processed. This is a DataContainerReference
State	A read/write python dictionary that is used to maintain state across function calls and iterations.

Only return values from BeforeIterationEx are processed. Returns from all other functions are discarded.

1. BeforeIteration(IterationNumber, ContainerList, State): This method is called before each iteration. Ideally used to initialize the state dictionary, open editors as required, or initialize setups as needed for each iteration.
2. BeforeIterationEx(IterationNumber, ContainerList, State): Similar to the BeforeIteration method except that this allows you to control the number of iterations via the return value.
   - Return: “more” to request one more iteration
   - Return “last” to indicate that this is the last iteration

- Any other return (including none) will be treated as a return of “last” and terminate iterations.

3. AfterIteration(IterationNumber, ContainerList, State): This method is called after each iteration. This can be used to copy result files over, check results, implement any possible convergence calculations, logging of results, etc.
4. BeforeContainerRefresh(IterationNumber, Container, ContainerList, State): Is called before each of the coupled containers is refreshed. The “Container” argument represents the container about to be refreshed.
5. AfterContainerRefresh(IterationNumber, Container, ContainerList, State): Is called after each of the coupled containers is refreshed. The “Container” argument represents the container just refreshed.
6. BeforeContainerUpdate(IterationNumber, Container, ContainerList, State): Is called before each of the coupled containers is Update (after a refresh). The “Container” argument represents the container about to be updated.
7. AfterContainerUpdate(IterationNumber, Container, ContainerList, State): Is called after each of the coupled containers is updated. The “Container” argument represents the container just updated.

If the callback scripts uses other files to send commands to various containers (vb, js, apdl, python, etc.), all of those files are best saved under the user_files directory. This allows you to use the FBGetUserFilePath(str) to get the absolute path of the file and allows the files to be packaged with any created archive.

Utility Functions

FBSystemForContainer (container)	system	Given a container, returns the system it belongs to.
FBSystemDisplayName (system)	string	Given a system, returns its display name on the schematic.
FBSystemID(system)	string	Given a system, returns its ID. This is the same as the UniqueDIrectory for the system.
FBGetUserFilesPath (relativePath)		Given a relative path located under the user_files directory, say "hello.py", this returns the absolute path of the file.

Output/Debugging Functions

FBAddInfoMessage(string)	Adds an info message to the WB message window.
FBAddWarningMessage (string)	Adds a warning message to the WB message window.
FBAddErrorMessage(string)	Adds an error message to the WB message window.
FBMessageBox(string)	Pops up a dialog with the supplied string with an “OK”

	button.

Example Scenarios for Feedback Iterator

This section describes several scenarios for using the Feedback Iterator.

Set up Iteration with Feedback Iterator

User Breaks Iteration Control

Start an Iterative Update

Run a Single Iteration

Interrupt an Iterative Loop

Resume an Interrupted Iterative Loop

Modify any of the Systems Involved in Iteration (coupled clients)

Iterating to Convergence

Set up Iteration with Feedback Iterator

Scenario	Steps to execute	Outcome
Electromagnetics product solution provides solution data to a single downstream setup	• Drop Feedback Iterator system on the Setup component of the Electromagnetics product OR • Create Feedback iterator system and connect Electromagnetics product setup component to the Feedback Iterator component AND User clicks on the Iterator component and sets the desired Feedback Iterator properties.	Electromagnetics product Setup component is connected to the Iterator. Electromagnetics product solution and downstream Setup and Solution components are coupled as clients to the iterator. The iterator component is the coupled master and the rest are coupled clients. Any coupling changes schematic visuals: • Coupled cells including coupling master are colored differently from normal cells. • Coupled cell icons change to reflect the icon of the coupling master. • Coupled client cells no longer display the **Update** context menu item and cannot be updated via script commands either. • The cell states of all coupled cells (master and clients) are synchronized to be the most pessimistic state among any of them. e.g., after an update, any modification will set all of them to the “Modified” state.
Electromagnetics	same as above	All the downstream components (Setup and

Scenario	Steps to execute	Outcome
product Solution provided to multiple downstream setup cells		Solution in each downstream system) are coupled as clients. The update order in this case depends on any additional data-flow connections between the systems that consume the upstream Electromagnetics product solution data.
Electromagnetics product Solution cell does not have any downstream targets	same as above	No changes occur beyond what is expected when you create a new connection (no coupling). Coupling only occurs when Electromagnetics product Solution cell has a downstream connection. If a downstream connection is added after the link to the iterator, coupling will be performed as described in the first row.

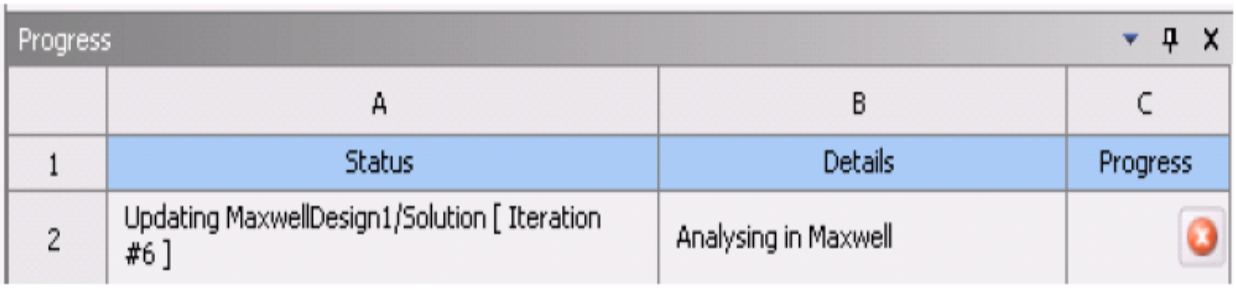
User Breaks Iteration Control

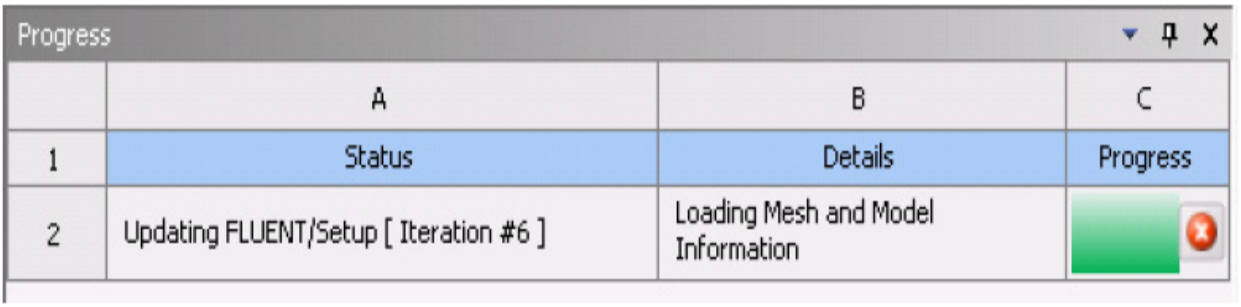
Scenario	Steps to execute	Outcome
Iterator coupled to Electromagnetics product solution cell and downstream setup/solution cells	• Break Electromagnetics product Solution cell's provides link -or- • Break Electromagnetics product Setup link to Feedback Iterator -or- • Delete Electromagnetics product Solution consumer system -or- • Delete Feedback Iterator system	In addition to what is normally expected from the user actions, all coupled clients are de-coupled (their icons and colors on UI are restored).
Electromagnetics product Setup cell is connected to the Feedback Iterator but the Electromagnetics product solution	• Break Electromagnetics product Setup link	No coupling exists in this scenario so nothing visible changes beyond

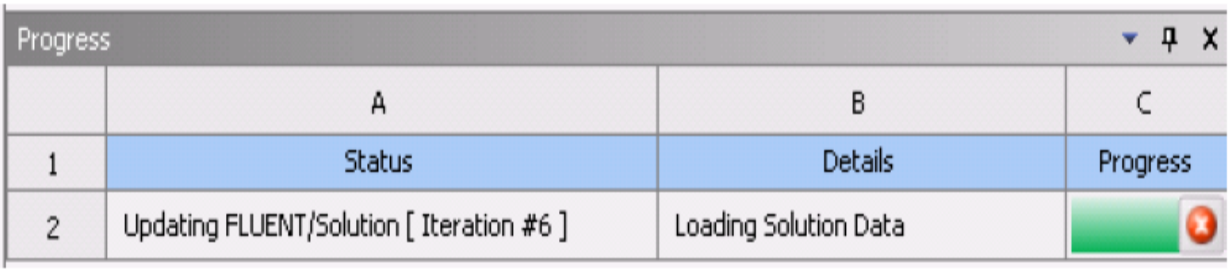
Scenario	Steps to execute	Outcome
does not provide anything. No components are coupled to the Feedback Iterator	to Feedback Iterator • Delete Feedback Iterator system	what is expected from the user actions.

Start an Iterative Update

Scenario	Steps to execute	Outcome
Standard update scenario	• Right-click on Feedback Iterator component and select Update operation -or- • Update Project -or- • Update all design points -or- • DX Update	For each iteration, the Electromagnetics product solution is updated followed by the downstream setup and the downstream solution. Each of the client component updates display their progress monitor and can be aborted (resulting in aborting the current iteration).







Run a Single Iteration

Scenario	Steps to execute	Outcome
NumCompleted < NumIterations	Right-click on Feedback Iterator component and select Single Iteration Update	One iteration is run. “Iterations Completed” property is incremented.
NumCompleted >= NumIterations	same as above	same as above

Interrupt an Iterative Loop

Currently, there is no special progress or interruption control. The progress monitor of the individual components is displayed as they are updated, and any control they choose to provide (interrupt, abort or both) are available. If the user chooses to abort any of the client component

updates, the current iteration is aborted and the Completed Iterations property remains unchanged from the previous iteration.

Resume an Interrupted Iterative Loop

Scenario	Steps to execute	Outcome
Single step	Right-click on Feedback Iterator component and select Single Iteration Update	One iteration is always run and the “Iterations Completed” property is incremented.
Run until completion	• Right-click on iterator component and select Update or- • Select the Project Update menu option from the toolbar etc.	If the user specified number of operations are already completed, nothing is done. Otherwise, the remaining iterations are run.

Modify any of the Systems Involved in Iteration (coupled clients)

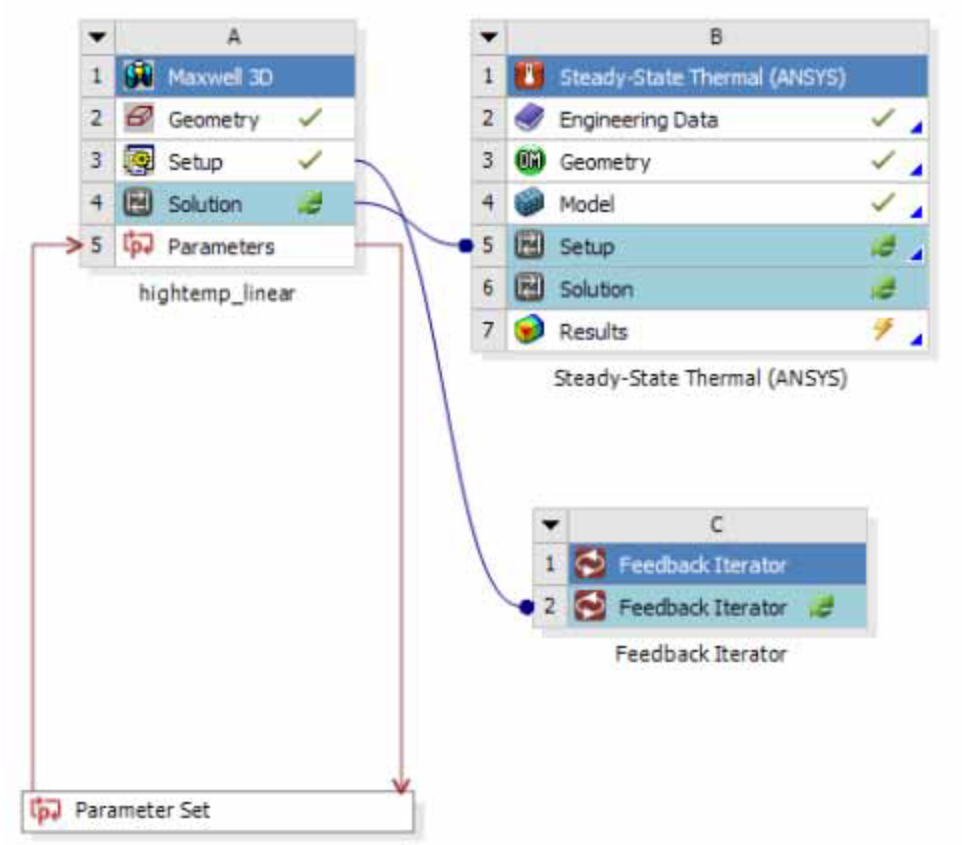
When the user modified any of the coupled client components either in Workbench by changing some non-extrinsic property, design point etc or in an external editor, the coupled clients and master will be marked as modified. Additionally, the Completed Iterations property will be set to 0.

Iterating to Convergence

Scenario	Steps to execute	Outcome
Standard update scenario	• Right-click on the Feedback Iterator component and select Update -or- • Update Project -or- • Update all design	For each iteration, the Electromagnetics product solution is updated followed by the downstream setup and the downstream solution. Each of the client component updates display their progress monitor and can be aborted (resulting in aborting the current iteration).

Scenario	Steps to execute	Outcome
	points -or- • DX Update	

The following temperature convergence example uses a Feedback Iterator with a Maxwell design coupled with a Steady-State Thermal component. Iterating to convergence operates similarly for HFSS projects.



The example uses defaults of 5% for the convergence targets and a value of 100 max iterations. Recall that the max iterations value is set as a "safety" measure to ensure that iterations do not continue indefinitely if the solution does not converge. As we progress through the iterations, we observe the following:

End of Iteration #1

The first iteration, since it has no previous iteration, cannot return a meaningful delta value.

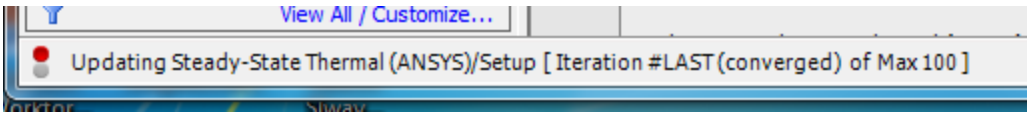
		started. (9:51:41 PM Jun 11, 2013)		
6	Informational	After iteration#: 1, Max temperature delta % = NaN, Max displacement delta % = NaN		6/11/2013 9:51:39 PM

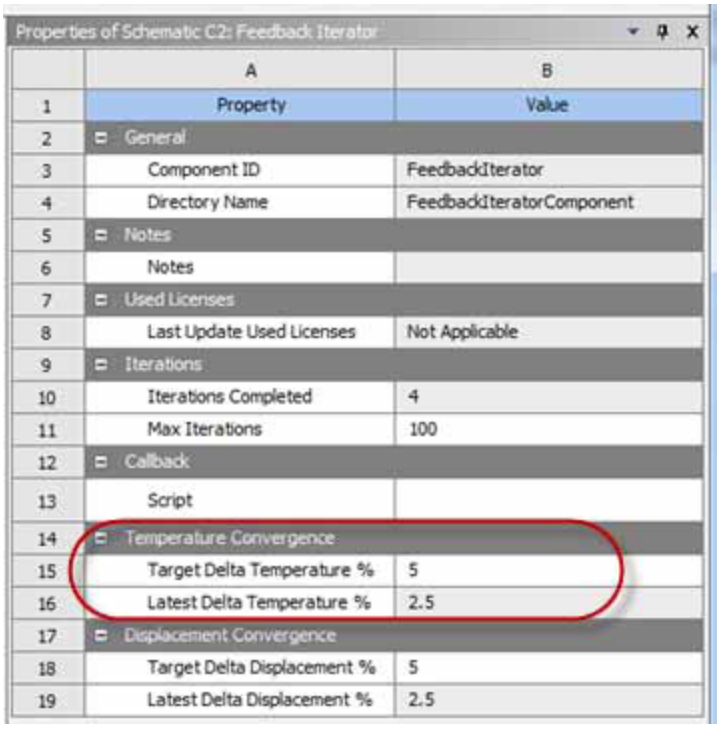
At the end of this iteration, the Latest Delta value will still be listed as Not Available.

	A	B
1	Property	Value
2	General	
3	Component ID	FeedbackIterator
4	Directory Name	FeedbackIteratorComponent
5	Notes	
6	Notes	
7	Used Licenses	
8	Last Update Used Licenses	
9	Iterations	
10	Iterations Completed	1
11	Max Iterations	100
12	Callback	
13	Script	
14	Temperature Convergence	
15	Target Delta Temperature %	5
16	Latest Delta Temperature %	Not Available
17	Displacement Convergence	
18	Target Delta Displacement %	5
19	Latest Delta Displacement %	Not Available

Convergence Achieved

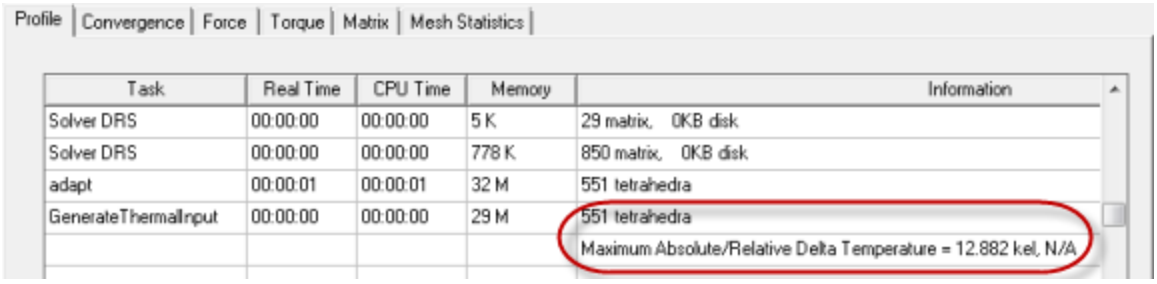
In this example, convergence has been achieved by the third iteration.





When the convergence criteria are met, the simulation stops. If you attempt to do an update after convergence has been achieved, the simulation will not be launched since the current solutions satisfy the convergence criteria.

You can also monitor the iteration progress by opening the Maxwell editor and from the Results menu, select the Solution data dialog box. Choose the Profile tab and keep the dialog open. As you solve each iteration, you can observe the reported Delta-T (with thermal feedback) and/or delta-displacement (with displacement feedback) and abort the iterations (or stop single iteration updates) when the values reach acceptable convergence levels.



19 - Exporting Equivalent Circuit Data

You can export lumped R, L, C data from a Maxwell solution to Ansys Twin Builder. Importing the new data file to Twin Builder enables you to include wave effects in the circuit simulations.

An equivalent circuit can be exported from a parametric solution or from an imported table.

Note	You can only export an equivalent circuit from a parametric solution when the following two criteria are met: • The solution type is **Magnetostatic** or **Electrostatic**. • A parametric setup exists.

Exporting a Circuit From a Parametric Solution

To create a circuit from parametric solutions, you first need to set up and solve a problem that contains a parametric sweep. Any force, torque, and matrix solutions are automatically available to use in the equivalent circuit. Other solutions can be treated as extra inputs/outputs if they have been added as calculations in the parametric setup — that is, extra inputs/outputs come from the parametric table.

Note	In Maxwell, global variable names begin with the **$** character. However, when you are creating an ECE **.sml** model, this character causes the circuit to fail when importing it to Twin Builder. To resolve this, when the circuit is exported, the **$** character is replaced by the **_** character. For example, the project variable **$MyVar** is converted to **_MyVar**.

To export a circuit from a parametric solution:

1. Click **Maxwell 3D>Export Equivalent Circuit>From Parametric Solutions** or **Maxwell 2D>Export Equivalent Circuit>From Parametric Solutions**.

   The **General** window appears, allowing you to specify basic information about the circuit model.

Note	This command is enabled only when the following two criteria are met: • The solution type is **Magnetostatic** or **Electrostatic**. • A parametric setup exists.

2. Select one of the following from the **Model Type** pull-down list depending upon the Design Type (2D or 3D):
   - **Linear Motion**
   - **Rotational Motion**
   - **Transformer**
   - **Matrix**
   - **Lookup Table**
3. Select a **Parametric Setup** from the pull-down list.
4. Select a **Solution Setup** from the pull-down list.
5. Select a **Matrix Setup** from the pull-down list.

6. Select a **Force Setup** or a **Torque Setup** from the pull-down list, depending on which parameter you have set up for your design.
7. For the force or torque setup, select **X**, **Y**, or **Z** as the **Component**.
8. Select either **Ampere-Turns** or **Amperes** as the **Current Source Variables** option.
9. Click **Next**.

   The **Table** window appears, allowing you to set up the inputs and outputs for the circuit equivalent. Most of the values have been automatically filled in, based on your design. You can keep the defaults or change the selections and values for the various parameters.
10. To apply deep spline interpolation to all inputs in the circuit PWL model, select the **Use Bezier Interpolation** check box.
11. To export the data to a table that you can edit and use to export an equivalent circuit at a later time, do the following:
    a. Click **Export Table**.

    The **Save As** dialog box appears.

    b. Select a location, and type a name in the **File name** box.
    c. Click **Save**.

    The **Table** window reappears. The file that is created contains header rows with information such as name, i/o, and type, plus all of the actual data. You may export the table to view or use in external programs. You can also modify the exported table file and then re-import it to create an equivalent circuit. On import, the information in the header rows is used as default settings in the circuit export dialog box.

Note	The **Table** window is the last step for the **Lookup Table** model type. For all other model types, there is one more step: the **Terminals** window.

12. For model types **Linear Motion**, **Rotational Motion**, **Transformer** and **Matrix**, click **Next**.

    The **Terminals** window appears, allowing you to set up terminals, which are nodes with "through" and "across" data. Most of the values have been automatically filled in, based on your design. You can keep the defaults or change the selections and values for the various parameters.
    1. Enter a **Scaling Factor** in the text box. The scaling factor is applied to all output quantities and can be used, for example, to scale data from partial models that take advantage of symmetry.
    2. Optionally, specify the **Model depth** for 2D XY models for scaling.
    3. Optionally for 2D XY **Rotational Motion** model type, specify the **Skew** angle.

       When a stator core or rotor core is skewed for a 2D Rotational Motion model, the flux linkages and torque of the look-up table will be modified based on the skew angle θ_k .

       For any output of the look-up table, $f(\theta)$, where θ is rotor position, the modified output will be:

$$f_m(\theta) = \frac{1}{\theta_k} \int_{\theta - \theta_k/2}^{\theta + \theta_k/2} f(\theta)\,\delta\theta$$

4. Specify the **Terminals**:
   - To specify a coil terminal:
     a. Select a **Source** for the **Flux** or **Charge**, depending on whether your design is mechanical or electrical.
     b. Enter a **Resistance**.
     c. Enter the number of **Turns** in the coil winding.
     d. Enter the number of **Branches**.
   - To specify a mechanical terminal:
     a. Select the **Force** or **Torque** variable from the pull-down list.
     b. Select the **Position** variable from the pull-down list.
     c. Optionally for Rotational Motion model type, select **Use rotational velocity**.
5. Click **Finish** to export the equivalent circuit.

Types of Equivalent Circuit Models

Currently, you can generate the following model types:

- **Linear Motion** - Models inductance (or capacitance) and force as functions of current (or voltage) and position. The parametric solution should cover the range of motion and conductor current (or voltage).
- **Rotational Motion** - Models using rotational rotor position sweep.
- **Transformer** - For one coil only, models nonlinear mutual (magnetizing) inductance with a linear leakage inductance branch. Therefore, this model is only valid for 1-phase transformers. The parametric solution should cover the range of current for one coil only in an open-circuit test. Nonlinear mutual inductances for 3-phase (or more) transformers are better solved using the transient solver.
- **Matrix** - Models an inductance or capacitance matrix for one or more conductors, as a function of current or voltage. The parametric solution should cover the range of each conductor's current or voltage.
- **Lookup Table** - This option creates a lookup table model in Simplorer format.

Exporting a Circuit From an Imported Table

To create a circuit from an imported table, you need to have a table file containing tabular data. The first row in the file must contain the column names. The table export feature writes other useful information into subsequent rows, but the import succeeds whether or not this information is present. The rows of tabular data come next.

Note	In Maxwell, global variable names begin with the **$** character. However, when you are

	creating an ECE **.sml** model, this character causes the circuit to fail when importing it to Twin Builder. To resolve this, when the circuit is exported, the **$** character is replaced by the **_** character. For example, the project variable **$MyVar** is converted to **_MyVar**.

To export a circuit from an imported table:

1. Click **Maxwell3D>Export Equivalent Circuit>From Imported Table or Maxwell2D>Export Equivalent Circuit>From Imported Table**.

   The **Select File** dialog box appears.
2. Find and select the file containing the table you want to use.
3. Click **Open**.

   The **General** window appears, allowing you to specify basic information about the circuit model.
4. Select one of the following from the **Model Type** pull-down list:
   - **Linear Motion**
   - **Rotational Motion**
   - **Matrix**
   - **Lookup Table**
5. Select either **Ampere-Turns** or **Amperes** as the **Current Source Variables** option.
6. Click **Next**.

   The **Table** window appears, allowing you to set up the inputs and outputs for the circuit equivalent. Most of the values have been automatically filled in, based on your design. You can keep the defaults or change the selections and values for the various parameters.
7. To apply deep spline interpolation to all inputs in the circuit PWL model, select the **Use Bezier Interpolation** check box.
8. To export the data to a table that you can edit and use to export an equivalent circuit at a later time, do the following:
   a. Click **Export Table**.

      The **Save As** dialog box appears.
   b. Select a location, and type a name in the **File name** box.
   c. Click **Save**.

      The **Table** window reappears. The file that is created contains header rows with information such as name, i/o, and type, plus all of the actual data. You may export the table to view or use in external programs. You can also modify the exported table file and then re-import it to create an equivalent circuit. On import, the information in the header rows is used as default settings in the circuit export dialog box.

Note	The **Table** window is the last step for the **Lookup Table** model type. For all other model types, there is one more step: the **Terminals** window.

9. For model types **Linear Motion**, **Rotational Motion**, and **Matrix**, click **Next**.

   The **Terminals** window appears, allowing you to set up terminals, which are nodes with “through” and “across” data. Most of the values have been automatically filled in, based on

your design. You can keep the defaults or change the selections and values for the various parameters.

1. Enter a **Scaling Factor** in the text box. The scaling factor is applied to all output quantities and can be used, for example, to scale data from partial models that take advantage of symmetry.
2. Optionally, specify the **Model depth** for 2D XY models for scaling.
3. Optionally for 2D XY **Rotational Motion** model type, specify the **Skew** angle.

   When a stator core or rotor core is skewed for a 2D Rotational Motion model, the flux linkages and torque of the look-up table will be modified based on the skew angle θ_k .

   For any output of the look-up table, $f(\theta)$, where θ is rotor position, the modified output will be:

$$f_m(\theta) = \frac{1}{\theta_k} \int_{\theta - \theta_k/2}^{\theta + \theta_k/2} f(\theta)\,\delta\theta$$

4. Specify the **Terminals**:
   - To specify a coil terminal:
     a. Select a **Source** for the **Flux** or **Charge**, depending on whether your design is mechanical or electrical.
     b. Enter a **Resistance**.
     c. Enter the number of **Turns** in the coil winding.
     d. Enter the number of **Branches**.
   - To specify a mechanical terminal:
     a. Select the **Force** (or **Torque)** variable from the pull-down list.
     b. Select the **Position** variable from the pull-down list.
     c. Optionally for Rotational Motion model type, select **Use rotational velocity**.
5. Click **Finish** to export the equivalent circuit.

Setting Up Current Variables

For problems with current, the **Current Variables Represent** setting allows you to specify one of the following two options:

- **Ampere-Turns**
- **Amperes**

This setting provides flexibility in setting up current sources with different numbers of turns and branches.

In Maxwell, current source values are specified in Ampere-turns, meaning that if a given source is driven with 2 Amps through 100 turns, then the source value must be set to 200. Inductance is calculated per-turn; therefore, to calculate EMF, the flux must be scaled by the number of turns to give the total flux.

You can create two kinds of circuit models:

- A model format where the data table is based on current in Amp-turns and flux per turn. This model internally converts currents and fluxes using the specified turn ratio. This model format was the only one available in versions of Maxwell before Maxwell 11.
- A model format where the data table is based on current in Amps and total flux. This model does *no* internal scaling of values, since the data is already in a form the outside circuit expects. This model format is available in Maxwell 11 and later versions.

For either model type, you can view the inductance setup to obtain information about the specified groups, turns, and branches.

Current Variables Represent Ampere-Turns

How to use this format:

- Apply variables directly to sources. *Source1* value = *Current1*.
- Sweep variable values in Ampere-turns. *Current1* sweeps from 100 to 200.
- In an inductance setup, specify the number of turns consistent with the source setup. Make sure all sources in a group have the same number of turns.

When to use this format:

- If all sources in a group have the same number of turns.
- If you want to be able to modify the number of turns in the exported circuit.
- If you are more comfortable thinking of variables in Ampere-turns.

When using this format, the grouping is taken from the inductance setup. The number of turns and branches per group from the inductance setup are used as a default but can be modified either before exporting the circuit or in the exported circuit.

Current Variables Represent Amperes

How to use this format:

- Define sources as the current variable times turns. *Source1* value = *100*Current1*.
- Sweep variable values in Amperes. *Current1* sweeps from 1 to 2.
- In an inductance setup, specify the number of turns consistent with the source setup.

When to use this format:

- In cases where one current is applied through sources with different numbers of turns.
- If you do *NOT* need to modify the number of turns in the exported circuit.
- If you are more comfortable thinking of variables in Amperes.

When using this format, the groups, turns, and branches are taken from the inductance setup and are used as is. You will *not* be able to modify them during circuit export or in the exported circuit.

Setting Inputs and Outputs in the Table Window

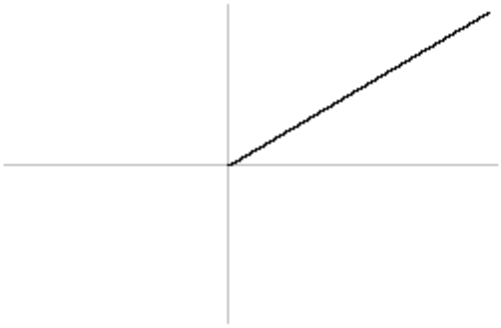
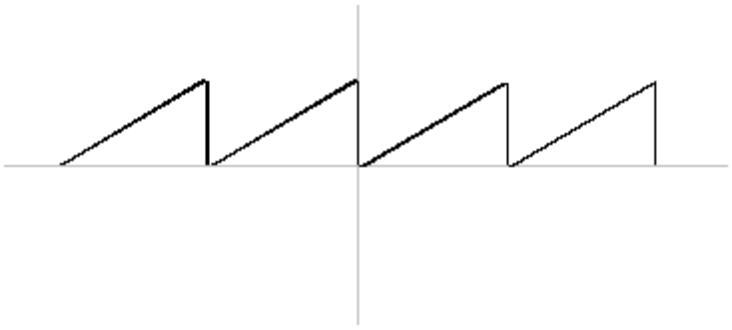
The table includes rows for all of the following that are appropriate to the model type:

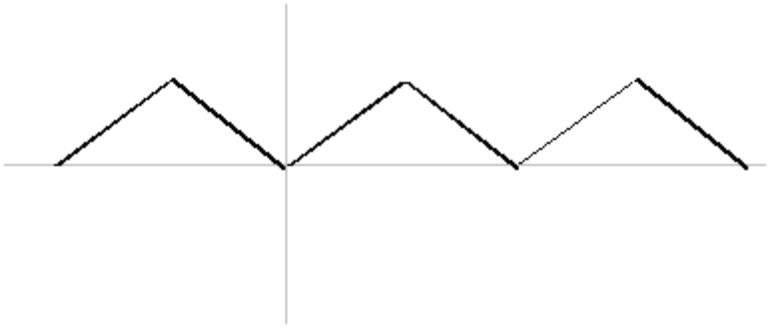
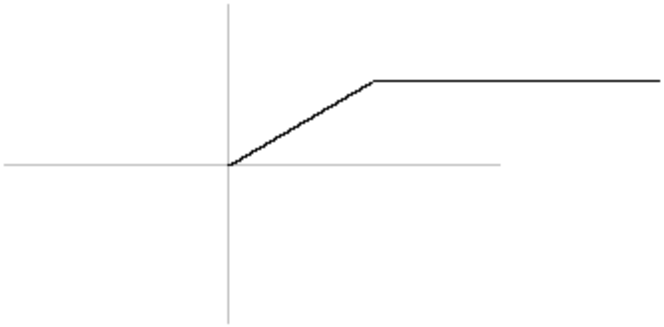
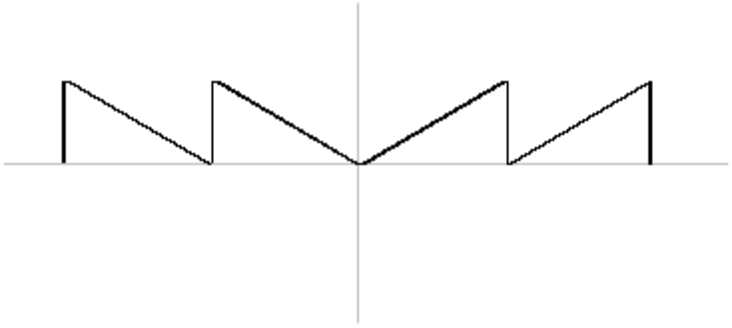
- Swept variables from the parametric setup.
- Components of flux or charge from the selected matrix.
- Force or torque.
- Calculations from the parametric setup.

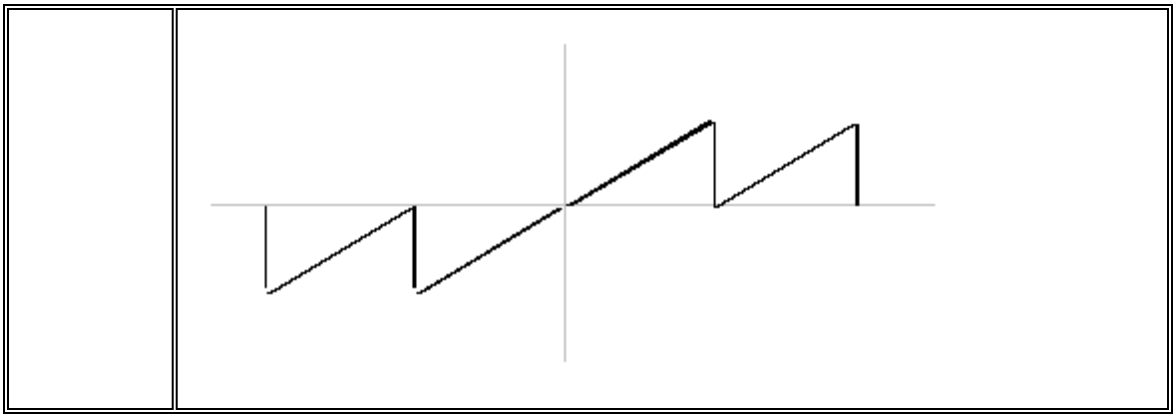
Where possible, the columns are set with reasonable defaults based on the information from the design. In many cases, you should not have to change anything.

Additional information about the columns:

- Under **I/O**, the choices are **Input**, **Output**, or **Unused**. **Unused** means that the quantity will not be included in the circuit.
- Under **Type**, the choices are **Position**, **Rotation**, **Current**, **Voltage**, **Flux**, **Charge**, **Force**, **Torque**, and **Other**. These choices affect the terminal setup in the next window. **Other** is typically used for extra inputs/outputs.
- Under **Extrapolate**, the choices are all standard extrapolations plus **None**. For inputs, the **Extrapolate** column is automatically set to **None** and is disabled.

Linear	This option takes the last two points of the data and generates a straight line extending beyond the range of the parametric sweep. 
Periodic	repeat the data outside the range of the parametric sweep. 
HalfPeriod	mirror then repeat the waveform outside the range of parameter sweep.

	
Constant	extrapolate a constant value from the last point in the interpolation. 
Even	repeat the wave outside the range of parametric sweep. 
Odd	repeat a reflection of the waveform outside the range of the parametric sweep.



Setting Coil and Mechanical Terminals

There are currently two types of terminals:

- Coil Terminals
- Mechanical Terminals

For models with motion, the mechanical terminal setup determines which force (or torque) causes the motion and which position (or rotation) is affected.

When setting up coil terminals, flux and current are used for magnetic models, and charge and voltage are used for electric models.

In the coil terminal setup, there is one row for each flux (or charge) group. In the **Current** (or **Voltage**) column, each cell is a selection box where you can choose the current (or voltage) that applies to that flux (or charge) group. Usually the software can choose the correct current by default, based on the source setup in the design. The **Resistance** column is set to **0** by default; you can enter any value here. The **Turns** and **Branches** columns only appear for magnetic models where current variables represent Ampere-turns. These columns are initialized with the turns and branches specified in the inductance matrix setup.

In the **Current** column, in addition to each current, there is another choice, **<Dependent>**. This feature allows you to solve some problems using fewer parametric rows. This is explained using the following example:

> Suppose you are working on a three phase machine. You create three sources, CurrentA, CurrentB, and CurrentC. In previous versions of Maxwell (before version 11), you would have to create three current variables, iA, iB, and iC, and sweep all three of these variables — even though for this type of machine iC is always equal to -(iA + iB). This means that if you want to sweep through 10 values of current, the parametric table would have 1000 rows. In Maxwell 11 and beyond, you can create only variables iA and iB and then set the value of CurrentC to
> -(iA + iB). The parametric table will have 10 values each for iA and iB, so it will only have 100 rows. When exporting the circuit, you would set the current for Flux[CurrentA] to iA, Flux[CurrentB] to iB, and Flux[CurrentC] to <Dependent>. In the circuit model, the PWL table will contain two input currents (iA and iB), but it will contain all three fluxes. So it will

look up all three flux values based on only the two current values. As long as you always connect a current of -(iA + iB) to CurrentC, this model will be valid.

Note	A dependent source can have a different number of turns from the sources that it depends on (in the example above, CurrentA, CurrentB, and CurrentC can all have different numbers of turns). But, in this case, you would have set up the current variables as Amperes instead of Ampere-turns — otherwise there would be no way to obtain the correct scaling of currents iA and iB for both their own current sources and for CurrentC. Therefore, this is another case where specifying current variables in Amperes (rather than in Ampere-turns) is useful.

20 - Scripting in Maxwell

Maxwell uses the Microsoft® Visual Basic® Scripting Edition (VBScript) scripting language to record macros. VBScript is based on the Microsoft Visual Basic programming language.

Using scripts is a fast, effective way to accomplish tasks you want to repeat.

You can write a script using any text editor, or you can record a script from within the Maxwell interface. After recording the script from within Maxwell, you can then modify it using a text editor.

For information on how to create, edit, and use scripts in Maxwell, see the Scripting help, which you can access from Maxwell via the following menu options:

- **Help >Maxwell Scripting Help**
- **Help >Maxwell PDFs >Maxwell Scripting**

Recording a Script

You can choose either to record a script to a file, or to record a script to the Project. Once you start to record a script, your subsequent actions are added to the script. Each interface command has one or more associated script commands that are recorded to the script.

To record a script to the project:

1. Click **Tools>Record Script to Project** to open the **Save Script to Project** dialog box.
2. In the **Save Script to Project** dialog box, type the name of the script and click **OK**.
3. Perform the steps that you want to record.
4. When you have finished recording the script, click **Tools>Stop Script Recording**. The script is recorded to a text file in .vbs (VBScript) file format and placed in the Project Manager Definitions>Scripts folder.

To record a script to a file:

1. Click **Tools>Record Script to File**.
   The **Save As** dialog box appears.
2. Use the file browser to locate and select the folder in which you want to save the script.
3. Type the name of the script in the **File name** text box and choose the file type (IronPython Script .py, or Visual Basic Script .vbs), and then click **Save** to save script in the folder you selected and begin recording.
4. Perform the steps that you want to record.
5. When you have finished recording the script, click **Tools>Stop Script Recording**.

Editing a Script

You can edit a script that was saved to a file, or edit a script that was saved to the project.

To edit a script that was saved to the project:

1. In the Project Manager Definitions>Scripts folder, right-click on the script you want to run to open the context menu.
2. Select **Edit Script** to open the script in the text editor.
3. When finished editing, select **Script> Save Script**.

To edit a script that was saved to a file:

1. Open the script file in any text editor.
2. When finished editing, save the script.

Stopping Script Recording

To stop recording a script:

- Click **Tools>Stop Script Recording**.
  Maxwell stops recording to the script.

Running a Script

To run a script from the Project Manager:

1. In the Project Manager Definitions>Scripts folder, right-click on the script you want to run to open the context menu.
2. Select **Run Script**.

To run a script from Maxwell:

1. Click **Tools>Run Script**.
   The **Open** dialog box appears.
2. Use the file browser to locate the folder in which you saved the script, and then double-click the folder's name.
3. Type the name of the script in the **File name** text box, or click its name, and then click **Open**.
   Maxwell executes the script.

Note	To supply script arguments when running from **Tools>Run Script**, use the edit field at the bottom of the file selection dialog box. You can access the script arguments using the AnsoftScriptHost.arguments collection from vbscript. This is a standard COM collection.

To run a script form a command line:

1. Type one of the following command line options at the command line:

   ```
   -runscriptandexit or -runscript
   ```

2. Press **Enter**.

You can also pass the **-scriptargs** parameter to the script and specify arguments.

Note	If you run the script from DOS prompt as a .vbs file (that is, you do not launch Maxwell, but simply launch vbs directly, or use wscript.exe or cscript.exe), the arguments will be in the WSH.arguments collection, not the AnsoftScriptHost.arguments collection. To

handle this, write the following:

```
on error resume next
dim args
Set args = AnsoftScript.arguments
if(IsEmpty(args)) then
Set args = WSH.arguments
End if
on error goto 0
'At this point, args has the arguments no matter if you are
running
'under windows script host or Ansoft script host
msgbox "Count is " & args.Count
for i = 0 to args.Count - 1
msgbox args(i)
next
```

Pausing and Resuming a Script

To pause a script during its execution:

- Click **Tools>Pause Script**.

To resume a script after pausing it:

- Click **Tools>Resume Script**.

Stopping a Script

To stop a script:

- Click **Tools>Stop Script**.

Maxwell stops executing the script that has been paused.

21 - Running Simulations

After you specify how Maxwell is to compute the solution, you need to begin the solution process. In general, the **Analyze** command applies to the selected setup and associated sweeps, if any, or to a select sweep. To use this command, right-click on a setup or sweep in the Project tree, and click the command on the context menu. The **Analyze All** command applies to all enabled setups at or below the level invoked in the Project tree. To use this command, either click Maxwell 2D or Maxwell 3D**>Analyze All** or right-click on the Analysis icon in the Project tree and select **Analyze All** from the popup menu.

What do you want to do?

- Local Analysis
- Solve a single setup with or without sweeps
- Solve a specific sweep
- Enable a queue so that multiple simulations can run sequentially as resources become available.
- Run more than one simulation, whether multiple setups, or multiple sweeps under a single setup, or setups with dependencies.
- Enable or Disable one or more solution setups or sweeps
- Monitor queued simulations
- Configure and run remote analysis
- Configure and run distributed analysis
- RSM Integration with Job Management UI
- Large Scale DSO for Parametric Analysis
- High Performance Computing (HPC) Integration
- Interactive Scheduler Jobs
- Monitor the solution process
- Change a solution priority for system resources
- Abort an analysis
- Re-solve after modifying a design
- Re-solve after Ansys Workbench Feedback

Related Topics

Running an Optimetrics Analysis

High Performance Computing (HPC) Integration

Running Maxwell from a Command Line

Technical Notes: *The Solution Process*

Solving a Single Setup

Running More Than One Simulation

Specifying Analysis Options

Remote Analysis

Monitoring the Solution Process

Aborting Analysis

Running an RMxprt Simulation

Local Analysis

Ansoft RSM Service does not need to be installed or running for local analysis and should not be running if the user only will be running local analysis.It is suggested that you refer to the Distributed Analysis page if running distributed analysis.

Note: MPI distribution may be affected by starting or stopping a VPN, even if the MPI distribution is limited to the local host.

To run more than one analysis at a time, follow the same procedure while a simulation is running. If you have enabled queuing, the next solution setup will be solved when the previous solution is complete.

Related Topics:

Distributed Analysis

Running Simulations

High Performance Computing (HPC) Integration

Running an Optimetrics Analysis

Disabling or Enabling an Analysis Setup

Solving a Single Setup

To solve a single setup or sweep:

1. In the project tree, under the design you want to solve, select a solution setup.
2. Right-click and select **Analyze** from the shortcut menu.

   Maxwell computes the 3D field solution inside the structure for a solution, and for a select sweep, does so for the sweep variables.

To run more than one analysis at a time, follow the same procedure while a simulation is running. If you have enabled queuing, the next solution setup will be solved when the previous solution is complete.

Note	If a linked dependency in the setup is already simulating (for example, due to setup links to the same external source for a near or far field wave, or a magnetic bias), Maxwell will not allow another dependent simulation to start until the first use of the source has completed.

Related Topics

Running an Optimetrics Analysis

Running More Than One Simulation

To solve every enabled solution setup in a design:

1. In the project tree, under the design you want to solve, select **Analysis**.
2. Click **Maxwell3D>Analyze All** or **Maxwell2D>Analyze All**.

Each enabled solution setup is solved in the order it appears in the project tree.

Note	The **General** tab for the Setup includes an **Enabled** check box. By default, this is checked. Unchecking the **Enabled** check box excludes a setup from running

Note	You can also simulate all designs in a project by clicking **Project>Analyze All**.

To solve *two or more sweeps or two or more parametric analyses under a setup*:

1. In the project tree, under the design you want to solve, right-click **the setup** icon that includes the setups or sweeps of interest.
2. Click **Analyze on the shortcut menu**.

   Each solution sweep under that setup is solved in the order it appears in the project tree, using the available machines.

Related Topics

Technical Notes: Handling Complicated Models

Solving a Single Setup

Specifying Analysis Options

Remote Analysis

Monitoring the Solution Process

Aborting Analyses

Running an Optimetrics Analysis

Monitoring Queued Simulations

If you have multiple setups for a design, and have selected **Analyze All**, the simulations can be queued until there is a machine available. You enable queuing in the HPC and Analysis Options: Options tab. If queuing is enabled and you run multiple setups, they are solved in the order that they appear in the project tree. You can prioritize setups by changing the order in the queue.

1. To view the solution queue, click **Tools >Show Queued Simulations**.

   This displays a dialog that displays each simulation and its current status. You select and remove any simulation from the queue.

   You can also select any setup and use the **Move up** and **Move down** buttons to prioritize them.
2. To remove a simulation from the queue, select the simulation, and click **Remove from Queue**.

This removes the selected simulation from the queue.

Monitoring the Solution Process

While a simulation is running, you can monitor the solution's progress in the **Progress** window. Above the green progress bar, messages describe the setup and step. The progress bar shows the relative progress of each step. Under the bar, messages note the part of the design being solved, and give memory estimates during the factoring process.

You can also view the following solution data at any time during or after the solution:

- The convergence data:
- The matrices computed for the S-parameters, impedances, and propagation constants.
- A profile of status of the adaptive analysis, including the number of valid passes completed.

To view the **Solutions** window:

1. Right-click the solution **Setup** in the project tree.

   A shortcut menu appears.

2. **Select Convergence**, **Solution**, **Matrix Data**, **Profile**, or **Mesh Statistics from the shortcut menu**.

   The **Solutions** window appears with the corresponding tab selected and the current data displayed.

For "out of core" problems, quite different amounts of memory may be used for factorization and for solution. So if the amount for factorization is displayed under the progress bar and the amount used is calculated for the profile at the end of the solution, they may be quite different numbers.

To view the status of the adaptive analysis:

- Click **Maxwell>Results>Browse Solutions**.

  The **Solutions** dialog box appears with the **Browse** tab selected. It displays data about the number of valid passes completed. It contains a tree structure showing the solutions listed according to Setup, Solution, and Variation. A table lists the Setup, the solution, the sweep variable, and the state of the solution.

- You can use the **Properties** button to display a dialog that lets you change the way the Setup, Solution, and Variation are listed in the tree structure of the **Solutions** dialog.
- The **Statistics** tab of the **Solutions** dialog displays path information, as well as format, number of files, and size.
- You can delete one or more solutions by selecting from the table and clicking **Delete**. Click on a solution to select it, and use Ctrl-click to select multiple solutions, or Shift-click to select a range of solutions. You can also select all solutions using the **Select All** button.

Note	If Maxwell loses its license, it waits for the license to be regained, checking every 2 minutes or until you abort.

Related Topics

Aborting Analyses

Deleting Solution Data

Changing a Solution Priority for System Resources

You can modify the priority of Maxwell simulations so that system resources are allocated to other computer processes before the solver. If you reduce the priority of Maxwell simulations, your other software tools will respond as they normally would, but Maxwell simulations may take longer.

Note	The Windows Task Manager does not indicate a reduced priority for the Maxwell solver. It only lists the priority of the engine manager, which appears normal, not the actual engine. The actual engine is in a separate thread, whose priority is not visible in the Windows Task Manager.

To change the priority of simulations for the system's resources:

1. While a solution is running, right-click the **Progress** window, and click **Change Priority** on the shortcut menu.
   - To affect priority for future simulation runs, click the **Tools>Options>HPC and Analysis** dialog box, and click the **Options** tab.
2. From the **Change Priority** menu (or the **Default Process Priority** pull-down menu), select one of the following priorities:

Critical	
Above Normal	
Normal	The default.
Below Normal	
Idle	

3. Click **OK**.

Related Topics

Monitoring Queued Simulations

Aborting Analyses

To end the solution process before it is complete:

- Right-click In the **Progress** window and click **Abort**.

Maxwell ends the analysis immediately.

The data for the currently solving pass or frequency point is deleted. All previously solved solutions are retained. For example, if you abort between the third and fourth adaptive pass, the solutions for the third pass will be available, and any solutions for the fourth pass are discarded.

To stop the solution process after the current adaptive pass or solved frequency point is complete:

- Right-click the **Progress** window, and click **CleanStop** on the shortcut menu.

  Maxwell ends the analysis after the next solved pass or frequency point.

If you request a clean stop between the third and fourth adaptive pass, the solutions for the third and fourth pass will be available once the fourth pass has finished solving.

Ansys Electromagnetics Application as an LSF Job

If you have an Ansys Electromagnetics application running as an LSF job, you can use the command “bkill –s SIGTERM *jobid*” to terminate an Ansys Electromagnetics application. Here *jobid* is the LSF job id. The response will be “Job <jobid> is being signaled”. The response is the same whether the job is actually being signaled or not.

In cases where the SIGTERM parameter is ignored, the command kills the LSF job, but does not clean the lock files, and other files may not be in a consistent state. See http://www.vital-it.ch/support/LSF/programmer/advanced.html for a detailed description under *Signal Handling in Windows*.

Unix/Linux

For UNIX/Linux, you can use TERM commands. Sigterm handling for Unix is done in Desktop library. You can abort a running batchsolve on Unix by sending a TERM signal to maxwell.exe

Related Topics

Integration with Platform’s Load Sharing Facility (LSF)

Re-solving a Problem

In some cases, if you modify a design after generating a solution, the solution in memory will no longer match the design. In such cases you receive a warning message that "Solutions have been invalidated. Undo to recover."

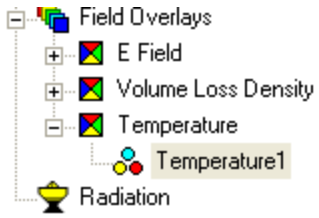
To generate a new solution after modifying a design, follow the procedure for running a simulation:

Also see Re-Solving with Ansys Workbench Thermal Feedback.

Re-solving after Ansys Workbench Thermal Feedback

With the Enable Feedback box in Setting the Temperature of Objects dialog is checked, you can manage analysis with feedback in Ansys Workbench. After solving a Maxwell design, after performing the corresponding linked thermal analysis in Ansys Workbench, you can receive a temperature distribution back from the thermal solution. Ansys Workbench will write the feedback files directly to the Maxwell Project Solution directory.

After an analysis that includes thermal feedback from Ansys Workbench, you can see temperature changes expressed in Temperature field overlays (both visually in the overlay and in the color key) as well as in the Solution data.



In the **Solution** data Profile tab you will see a new entry for Maximum Delta T, for the change in temperature from the previous simulation. The solver calculates delta in the first iteration by comparing the temperature distribution output from thermal with the initial temperature setting in Maxwell. Subsequent simulation iterations provide a number for the temperature delta.

Solver MCS4	00:00:00	00:00:00	31.4 M	Disk = 0 KBytes, matrix size 3468 , matrix bandwidth
Field Recovery	00:00:00	00:00:00	31.4 M	Disk = 310 KBytes, 2 excitations
				Maximum Delta T = 7.0685

This simulation feedback loop from Ansys Electromagnetics to Ansys Workbench and back can continue until you decide that Temperature delta reported in the Solution Report low and stable for the designs.

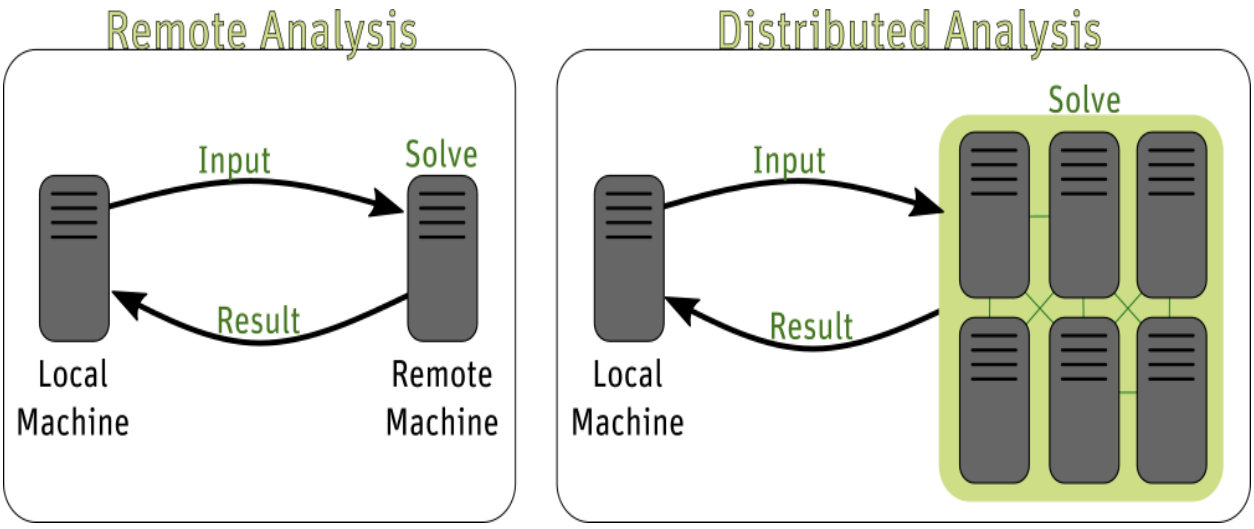
Related Topics

Setting the Temperature of Objects

Ansys Workbench Integration Overview

Remote Analysis

It is possible to solve a project on a different machine from the one on which you set up your designs. This is particularly useful when you want to take advantage of a more powerful machine but it is not convenient to access that machine. This process involves configuring the machine that is to perform the solving (the remote machine), as well as the machine from which the simulation is to be launched (the local machine). This can also be extended into distributed analysis, where a specified analysis, if supported, is concurrently solved on multiple remote machines.



Note	In both Remote and Distributed Analysis, communication between machines can drastically affect performance. Use of a high-speed network system, such as Gigabit or Infiniband, is recommended for optimal performance.

- Prerequisites for Remote and Distributed Analysis
- Configuring the Local Machine to Solve Remotely
- Remote Analysis Options
- Running Remote Analysis

The Tools > Options > Export Options Files command writes xml files containing the Options settings at all levels to the specified directory. The **Tools > Options > Export Options** feature is intended to make it easier for different users to use Ansys Electromagnetics tools installed on shared directories or network drives. The Example Uses for Export Options Features section outlines some use cases enabled by this feature.

Prerequisites for Remote and Distributed Analysis

1. You must have Ansys Electromagnetics Remote Simulation Manager (RSM) or a supported High Performance Computing (HPC) management software program. (See High Performance Computing (HPC) Integration). The list of currently-supported HPC software includes
   - Platform's Load Sharing Facility or LSF
   - Altair's PBS
   - GridEngine
   - Microsoft® Windows® Compute Cluster Server 2003
   - Microsoft® Windows® HPC Server 2008 R2 SP4, HPC Server 2012
2. All types of distribution use MPI, except when only distributing rows of a parametric table, either as a regular DSO or as Large Scale DSO performed through command line. MPI may also be used if Auto mode is specified, and rows of a parametric table are distributed. MPI must be correctly configured if the distributed analysis uses MPI. See Distributed Analysis

for information on whether MPI is needed for the analysis.

3. If you use RSM, it must be accessible from all remote machines. In addition, the Maxwell engines must be registered with each initialization of RSM. To do this, on each remote machine:
   - On Windows on the local and remote machines, click **Start > All Programs > Ansys Electromagnetics Suite [version] > Register with RSM**. You can also run **RegisterEnginesWith RSM.exe**, located in the product subdirectory (for example, C:\Program Files\AnsysEM\v231\Win64\RegisterEnginesWithRSM.exe).

   In each case, you see a dialog confirming the registration. OK the dialog.

   - On Linux, run **RegisterEnginesWithRSM.pl**, located in the product installation directory. (for example, /apps/AnsysEM/v231/RegisterEnginesWithRSM.pl).

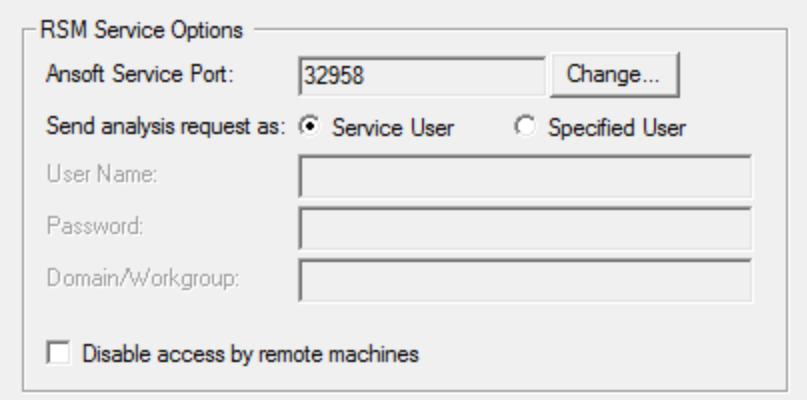
If the RSM service cannot run due to permission issues for the configuration file, it issues an error message and exits. If your product is not registered with RSM, the analysis will run locally.

Configuring the Local Machine to Solve Remotely

To set the Analysis options in Maxwell see Configuring Distributed Analysis.

Remote Analysis Options

You also set the Remote Analysis Options in the **General options** dialog box, **RSM Analysis Options** tab.



Select whether to run simulation processes as the user running RSM (Service User), or a Specified User. If you select Specified user, you must provide the User Name, Password, and any Domain/Workgroup on which this user is defined. If the name or password is incorrect, the Message window issues a warning message, and the solver attempts to perform the analysis as the Service User.

Running Remote Analysis

When you run a simulation remotely, you should see a message in the Progress window identifying the design name, and the specified remote machine. You will see Progress messages

as the simulations continues. When the simulation is complete, you will see a message in the **Message** window.

Related Topics

Setting Analysis Options
Distributed Analysis
Troubleshooting
Running from a Windows Remote Terminal

Troubleshooting

Problem: When you try to solve from local to remote machine, an MAXWELLCOMENGINE process starts on the remote machine, but the Maxwell user interface hangs indefinitely.

This occurs if the remote solve option is enabled after the COM daemon is started, or when the option "Don't allow exceptions" is selected for the Windows firewall.

Resolution: Remote solve needs either firewall exceptions to be ON or firewall to be completely turned off.

Problem: When you try to solve from a local to a remote machine, you receive the following error message:

[error] Unable to locate or start COM engine on 'nomachine' : Unable to reach AnsoftRSMService. Check if the service is running and if the firewall allows communication. (10:57:13 PM Aug 13, 2009)

Resolution: This message can happen if the machine is not present, the network connection is down, if there are firewall issues or if the service is not running.

Problem: A solve that is distributed to multiple hosts using MPI fails because the AnsysEM installation path is different on different hosts.

Resolution: Ansys Electronics Desktop must be accessible from all remote machines as well as on the local machine. If the analysis uses MPI, then the path of the Ansys Electronics installation must be the same on all of the machines used for the analysis (remote and local). This may be a shared network path accessible from all hosts. Alternatively, it may be a local installation on each host; in this case the installation path must be the same for all hosts. If using a shared network path, there should not be a local installation of the same Ansys Electronics version on any of the hosts. See Distributed Analysis for information on whether MPI is needed for the analysis.

Remote Solve Node = Windows

Error: "Unable to locate or start COM engine on <remote node> : Unable to reach AnsoftRSMService. Check if the service is running and if the firewall allows communication."

Resolution:

1. Try disabling the firewall.
2. Confirm that you have not changed the Ansoft Service Port in **Tools > Options > General Options > Remote Analysis Options** from the default 32958. If you have, change it back to 32958, restart Maxwell, and try to solve again.
3. Make sure that the local machine is able to contact the RSM port on the remote node. Open a command prompt on the local machine and type telnet <remote node name> 32958. If the terminal appears to be hanging then the connection was successful.
4. Make sure the user listed in the service is an administrator.
5. Make sure the COM engine is registered with the Ansys Electromagnetics RSM Service. From the Windows menu, choose **Start>All Programs>Ansys Electromagnetics>Ansys EM Suite [version]>Register with RSM** to register the engines.
6. If none of these steps fixes the problem, contact Ansys Support.

Error: "Unable to locate or start COM engine on <remote node>: Engine is not registered with the Ansys Electromagnetics RSM service which is running on this machine."

- **Resolution:** To register the engine, from the Windows menu, select **Start>All Programs>Ansys Electromagnetics>Ansys Electromagnetics Suite [version]>Register with RSM**.

Remote Solve Node = Linux

Error: "Unable to locate or start COM engine on <remote node>: Unable to reach AnsoftRSMService. Check if the service is running and if the firewall allows communication."

Resolution:

1. Try disabling the firewall.
2. Confirm that you have not changed the Ansoft Service Port in **Tools >Options >General Options >Remote Analysis Options** from the default 32958. If you have, change it back to 32958, restart the Ansys Electromagnetics product, and try to solve again.
3. Make sure that the local machine is able to contact the RSM port on the remote node. Open a command prompt on the local machine and type telnet <remote node name> 32958. If the terminal appears to be hanging then the connection was successful.
4. Check to make sure Remote Simulation Manager is running. To do this:
   a. Go to the 'rsm' subdirectory of the Ansys Electromagnetics Remote Simulation Manager installation directory, <RSM installdir>/rsm.
   b. Type `./ansoftrsmservice status`
   c. If the status query indicates that the service is stopped, type `./ansoftrsmservice start`
5. Make sure the COM engine is registered with RSM. Type `./RegisterEnginesWithRSM.pl status` from within the Maxwell installation directory. If the status query indicates "Not registered", type:

   ```
   ./RegisterEnginesWithRSM.pl add
   ```

Error: "Unable to locate or start COM engine on <remote node>: Engine is not registered with the Ansys Electromagnetics RSM service which is running on this machine."

Resolution:

- To register the engine, go to the Ansys Electromagnetics product installation directory and type:

  ```
  ./RegisterEnginesWithRSM.pl add
  ```

Distributed Analysis

Distributed analysis allows users to split certain types of analyses and solve each portion of an analysis simultaneously on multiple machines. Simulation times can be greatly decreased by using this feature.

> **Note:** When using distributed analysis, the Temp folder must be accessible by all machines set up in the HPC configuration.

Maxwell supports the following forms of distributed analysis:

- Distributing rows of a parametric table, either as a regular DSO, or as Large Scale DSO performed through command line. Large Scale DSO generates a reduced set of outputs.
- Distributing a single or discrete interpolating sweep.
- Maxwell 2D and Maxwell 3D designs support distribution of frequencies, transient solver, solution matrix and skew model.

> **Note:** Communication between machines in remote analysis and distributed analysis can drastically affect performance. Use of a high-speed network system, like Gigabit or Infiniband, is recommended for optimal performance.

Maxwell Circuit and RMxprt designs do not support MPI. As a result, the supported forms of distribution depend on the design type.

All types of distribution for Maxwell 2D and Maxwell 3D use MPI, except when only distributing rows of a parametric table, either as a regular DSO or as Large Scale DSO performed through command line. MPI may also be used if Auto mode is specified, and rows of a parametric table are distributed. MPI must be correctly configured if the distributed analysis uses MPI. See Setting HPC and Analysis Options for setting MPI Licensing and the *Ansys Electronics Desktop Installation Guide* for details on installing MPI.

To configure a distributed analysis, you must select a distributed machine configuration containing a list of machines to use for a simulation, based on memory and CPU considerations (See: Selecting an Optimal Configuration for Distributed Analysis). To create an new distributed machine configuration or to edit an existing one, see Editing Distributed Machine Configurations.

Before you can select a configuration, it must be active. See: Setting HPC and Analysis Options.

To select an existing, active configuration:

- From the **Simulation** tab, use the **Active** drop-down menu to select a configuration.

Related Topics

Configuring Distributed Analysis

Editing Distributed Machine Configurations

Selecting an Optimal Configuration for Distributed Analysis

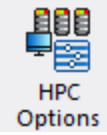
Large Scale DSO for Parametric Analysis

Configuring Distributed Analysis

To configure distributed analysis you select a distributed machine configuration. This is a list of machines to use for a simulation, based on considerations such as whether the simulation is more memory intensive or more CPU intensive, relative to the resources available on your network. (See Selecting an Optimal Configuration for Distributed Analysis for a discussion of issues.) To create an new distributed machine configuration. or to edit an existing one, see Editing Distributed Machine Configurations.

To select from an existing configuration:

1. Click the **HPC Options** icon on the **Simulation** ribbon,

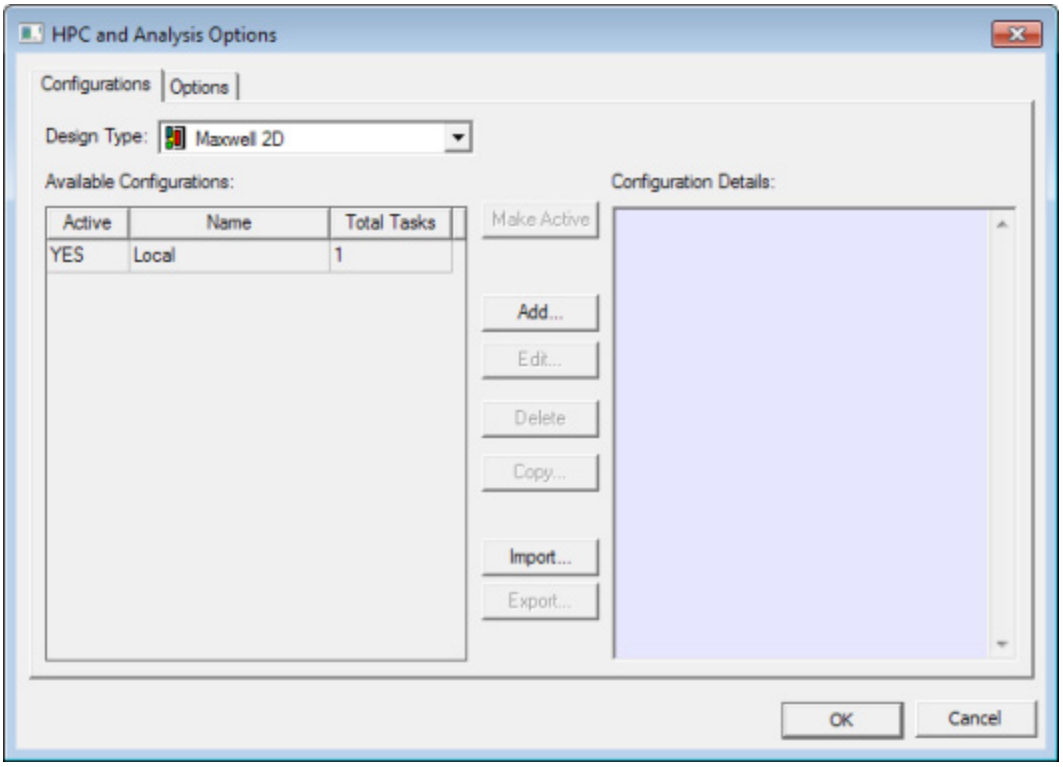


or click **Tools>Options>HPC and Analysis Options.**

This opens the **HPC and Analysis Options** dialog box. You can define and select configurations for Maxwell 2D, 3D, and RMxprt design types separately. You can view a list of Available configurations, and a report of the configuration details. From the current, list, you can select a configuration to **Make Active**. You can also **Add** a new configuration, **Edit** an existing one, or **Export** as a Ansys Configuration file (*.acf). You can also **Import** a configuration file. This lists existing configurations, and shows all machines in the selected configuration, enabled or not. You can **Copy** an existing configuration, typically to edit the name and contents for other purposes.

In the **Options** tab you can enable queuing, set the design type, set licensing options and specify default process priority.

For a more detailed discussion of this dialog, see Setting HPC and Analysis Options.



2. To define a new configuration, on the **Analysis Settings** dialog box, click the **Add** button to open the **Analysis Configuration** dialog box. See Editing Distributed Machine Configurations.

Related Topics

Editing Distributed Machine Configurations

Selecting an Optimal Configuration for Distributed Analysis

Setting Options in Maxwell

Setting HPC and Analysis Options

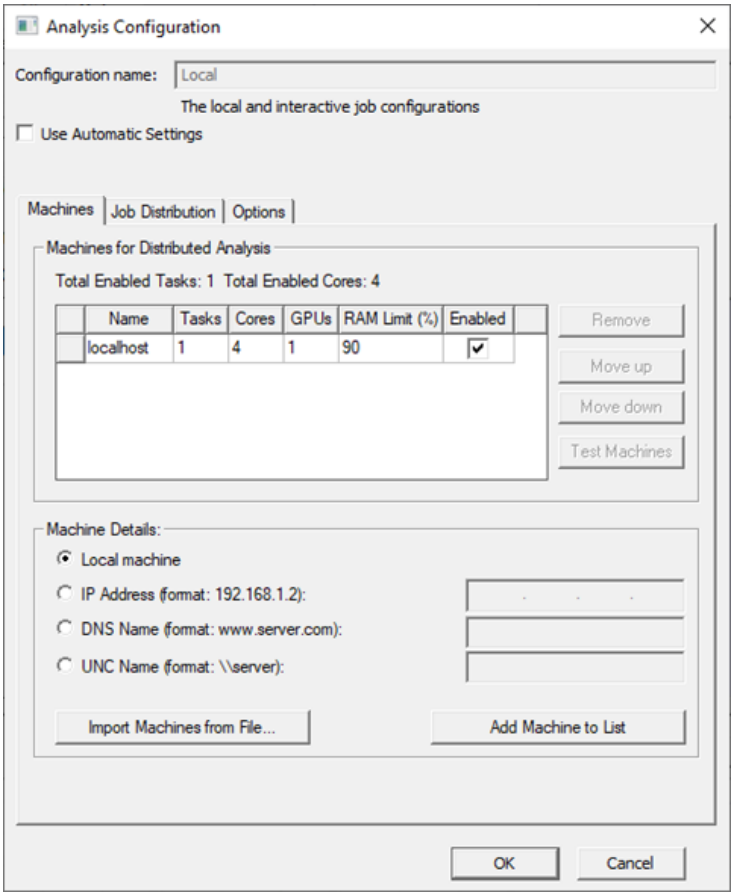
Editing Distributed Machine Configurations

To edit an existing machine configuration:

1. Click **Tools>Edit Active Analysis Configuration** to open the **Analysis Configuration** dialog directly or click the **Analysis Config** icon on the Simulation ribbon.

   This opens the **Analysis Configuration** dialog directly. You can also access this dialog from the **HPC and Analysis Options** dialog by clicking the **Add..,**, **Edit...,** or **Copy...** buttons.

If you select **Add...** from the **HPC and Analysis Options** dialog box, the fields are empty. If you select **Edit...** or use **Tools>Edit Active Analysis Configuration**, or click **Copy** from the **HPC and Analysis Options** dialog box, the fields show the selected configuration.



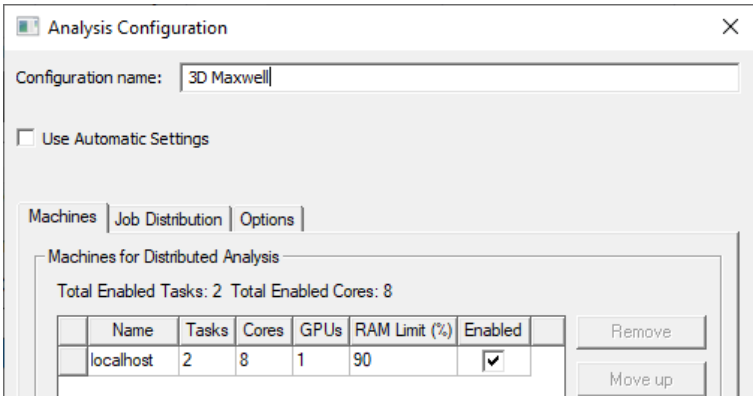
The option and features available differ for the various solvers.

Note	The **GPUs** column does not appear for Maxwell 2D solves. Use of GPUs is not currently supported for Maxwell 2D.

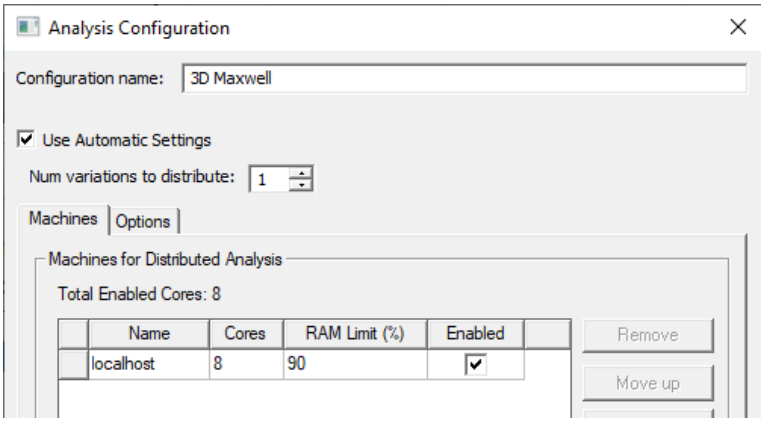
2. If you **Add** a new configuration, you must specify the name of the new or edited configuration. It cannot be empty and cannot be a previously used name or a reserved word.

3. If the solver supports automatic settings, you can select the **Use Automatic Settings** check-box to assign resources automatically, or you can uncheck the box to specify job distribution manually in the **Job Distribution** tab, as discussed below. For Maxwell, this

setting currently is supported only for 2D and 3D Transient, and 2D and 3D Eddy Current designs.



Selecting **Use Automatic Settings** removes the **Job Distribution** tab, setting those parameters automatically based on the best use of available resources for the current analysis.

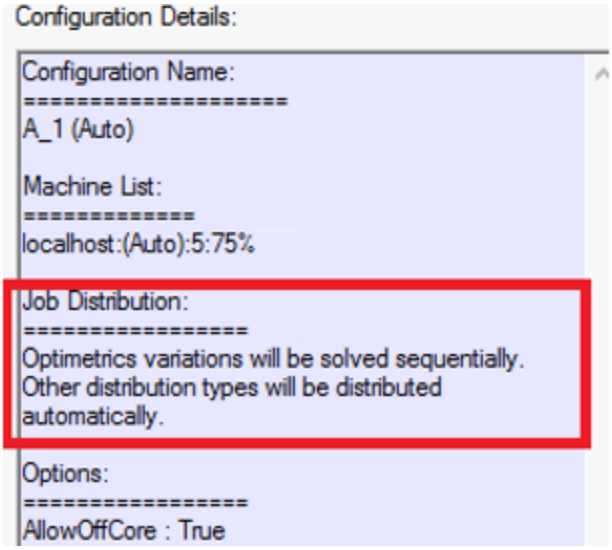


For Maxwell 2D and Maxwell 3D designs **Use Automatic Settings** must be checked to enable **Num variations to distribute**. In such cases, if you specify more than 1, the distribution of variations is in proportion to RAM availability on the machines listed. Depending on the resources available, and whether the simulation defines Optimetrics variations and other distributed types, the simulation can include up to three levels of distribution. When you close the **Analysis Configuration** dialog box, the **HPC and Analysis Options** dialog **Configuration Details** field provides details of the Job Distribution for 3 level distribution.

For 2D and 3D Transient designs, the **Use Automatic Settings** option distributes one excitation to each machine listed in the configuration. When this option is enabled, a minimum of 2 cores must be assigned. Each machine uses all cores assigned to it while solving its excitation. Unlike automatic frequency setup, the machines are assigned

sequentially without regard to machine capability. For 2D and 3D Eddy Current designs with **Use Automatic Settings**, a minimum of 4 cores must be assigned.

Variations will not be distributed when the number of variations to distribute is set to 1. In this case when you close the **Analysis Configuration** dialog box, the **HPC and Analysis Options** dialog box, **Configuration** tab provides details of the Job Distribution that does not use 3 level distribution.



4. The **Machines Tab** contains the machine list for the analysis configuration.

   Here you can provide machine information, either by specifying Machine Details, or by importing a list of machines from a file. You can then remove, order, test, and enable machines on the list. Control buttons let you **Add machine to List** or **Remove** machines from the list.

   Numbers of Tasks and Cores

   If automatic settings are either not used or not available, note that each machine has an associated number of tasks and number of cores. The number of tasks specifies the total number of compute jobs that will be run on that machine simultaneously. Each separate solver, or distribution is one task. The total cores specifies the total number of cores that will be used on the given machine. This is how you specify multiprocessing. For example, if you want to run two threads for each task, you specify Total Cores = 2 x Number of Tasks. The number of cores must always be greater than or equal to the number of tasks. If the number of cores is not an exact multiple of the number of tasks, some tasks will use more cores than others. For example, if Number of Tasks is 4 and Total Cores is 10, 2 tasks will use 3 cores, and 2 tasks will use 2 cores.

   GPUs

Note	• The **GPUs** column does not appear for Maxwell 2D solves. Use of GPUs is not currently supported for Maxwell 2D.

	• The Electronics Desktop supports GPU acceleration only for Maxwell 3D Eddy Current Solution Matrix solves.

To enable use of GPU acceleration for Maxwell 3D solves, you set **Enable GPU** to **True** on the **Options** tab in the **HPC and Analysis Options** dialog (see Setting HPC and Analysis Options); and disable (uncheck) **Use Automatic Settings** in the **Analysis Configuration** dialog box. The matrix solver automatically determines if all cores should be used, or if one GPU should be used to give the best performance. For example, for a Maxwell 3D Eddy Current Solution Matrix distribution, if you specify 4 cores for the simulation, the 3D eddy current solver will use 4 cores in parallel during matrix assembly while the matrix solver will use either 4 cores or 1 GPU.

RAM Limit (%)

Here you can specify the maximum percentage of the machine's RAM to be used for processing.

Import Machines from File...

You can import a machine list from a file, and an enhanced the file format handles the new flexibility. Each line of the file can contain a machine specifier of the form:

<MachineName>:<NumTasks>:<NumCores>

Note: this same format is used with the "-machinelist file=" command line option.

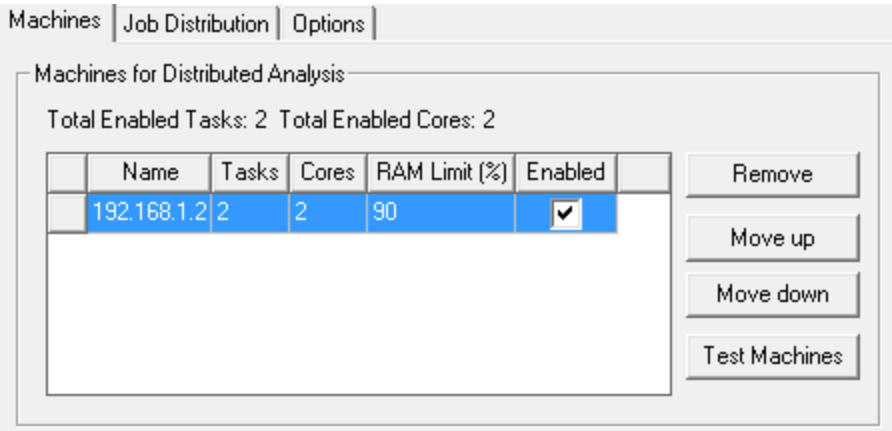
Local Machine Radio Button

To streamline the common case of running jobs on the current machine, use the dedicated radio button to specify the local machine.

5. For each machine to manually add to the list, under Remote Machine Details, specify an IP address, a DNS name, or a UNC name.

   The remote machines must have the same ANSYS Electromagnetics Suite version installed in the same OS and version, and have the RSM service active.

   Once you have specified the remote machine details, either directly or by **Importing Machine from a File**, you select a machine from the table to enable the buttons to **Remove** a machine, or to **Move** a machine up or down on the list.



The displayed list shows the order in which you entered them irrespective of the load on the machines. If you have selected **Use Automatic Settings**, the Tasks column field is disabled because you no longer need to make those assignments. To control the list order, select one or more machines, and use the **Move up** or **Move down** buttons. **Move up** and **Move down** are enabled when you select one or more adjacent machine names.

Enabled Machines

Each machine on the current list has an **Enabled** checkbox. Here you can enable or disable the listed machines according to circumstance. Above the table, the dialog gives a count of the total enabled tasks, and the total enabled cores.

For distributed tasks, the software will allocate the total cores on a given machine to that machine's tasks. If a machine with 8 cores is running 2 distributed tasks, the software will automatically allocate 4 cores to each task. If it is running 4 distributed tasks, each gets 2 cores. And if it is running 3 distributed tasks, the first two tasks get 3 cores and the last task gets 2 cores.

For a given variation (for example, frequency or geometry), you should make assignments so that each task has the same number of cores. This is because the solvers attempt to make each task computationally balanced. For example, with two machines, one with eight cores, and another with four, assuming that the memory is proportionally equivalent, you could assign two tasks for machine 1, and one task for machine 2, giving all tasks the same number of cores.

In general, Ansys Electromagnetics Suite solvers use machines in the distributed analysis machines list in the order in which they appear. If you select a distributed configuration (rather than Local) on the ribbon Simulation tab, and you launch multiple analyses from the same UI, Ansys Electromagnetics Suite solvers select the machines that are running the fewest number of distributions in the order in which the machines appear in the list. For example, if the list contains 4 machines, and you launch a simulation that requires one machine, the solver chooses the first machine in the list. If another simulation is launched while the previous one is running, and this simulation requires two machines, the solver

chooses machines 2 and 3 from the list. If the first simulation then terminates and we launch another simulation requiring three machines, the solver chooses 1, 4, and 2 (in that order)

Test Machines

Test Machines- When multiple users on a network are using distributed solve or remote solve, they should check the status of their machines before launching a simulation to ensure no other Ansys Electromagnetics processes are running on the machine. To do this, you can select one or more machines and click the **Test Machine**s button. A **Test Machines** dialog opens.

The test goes through the current machine list and gives a report on the status of each machine. A progress bar shows how far testing has gone. An Abort button lets you cancel a test. When the test is complete, you can Close the dialog box. If you need to disable or Enable machines from the list based on the report, you can do so in the **Distributed Analysis Machines** dialog.

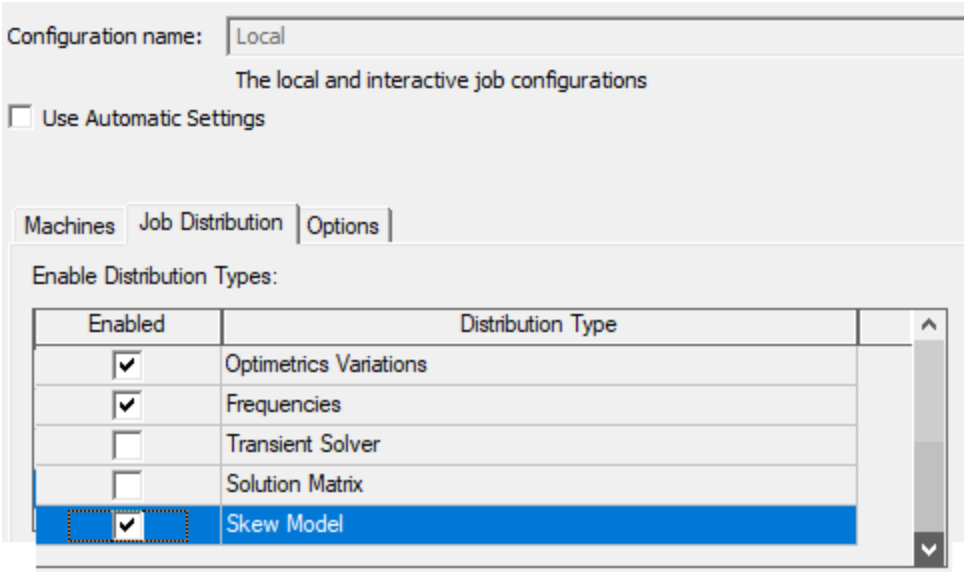
Job Distribution Tab

Use this tab to manually enable specific job distribution types, and to enable multilevel solves. The **Job distribution** tab is disabled if you select **Use Automatic Settings**. Use this tab to manually enable specific types of job distribution and to enable multi-level solves.

Job distribution types are design type specific and will differ between solvers.

- Maxwell 2D and 3D design types can use **Optimetrics Variations**, **Frequencies**, **Transient Solver**, **Solution Matrix**, and **Skew Model** distribution types.

- RMxprt can use only **Optimetrics Variations**.



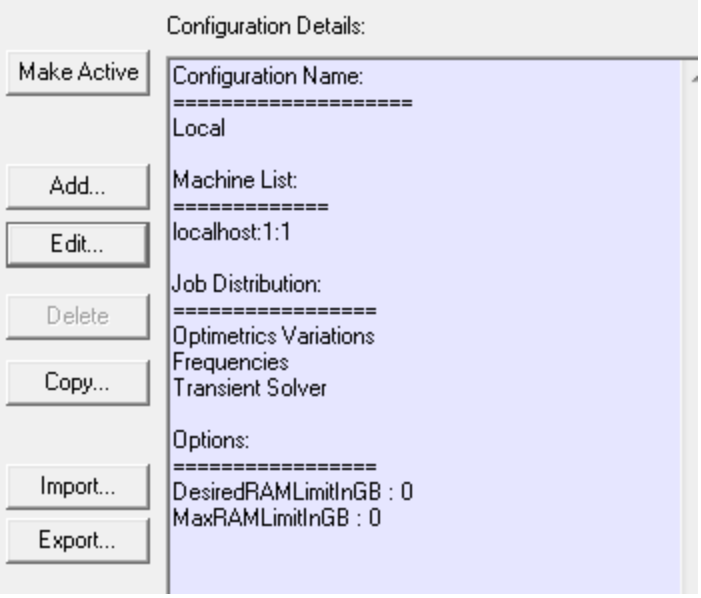
Use the check boxes to enable/disable available distribution types. The job distribution list box allows you to specify which job distribution types to allow for the current analysis configuration. At solve time, the Ansys Electromagnetics software automatically selects the best distribution type from the enabled distribution types. By enabling/disabling distribution types, you can control the job distribution.

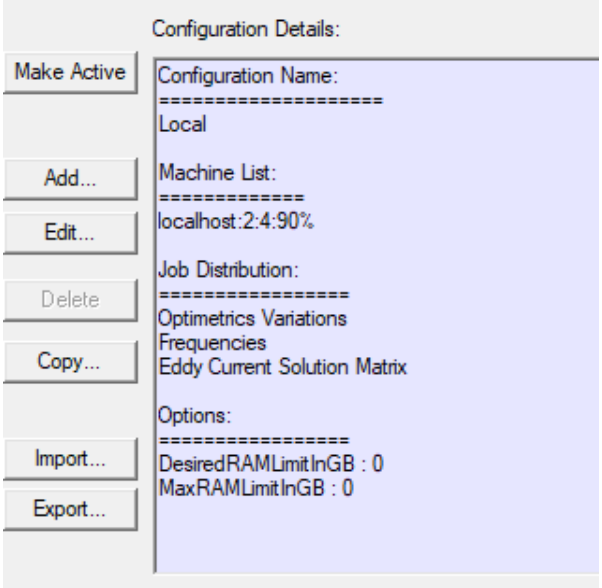
Enabling a distribution type does not necessarily mean it will be used. It must also be allowed by the solve setup. For example:

- the **Transient Solver** distribution type that enables use of the Time Decomposition Method, will only be *used* with Maxwell 2D and Maxwell 3D transient designs.
- The **Solution Matrix** distribution type that enables use of the Matrix-based Domain Decomposition Method will be used with both 3D eddy current designs and 3D magnetostatic designs.
- The **Skew Model** distribution type for use with Maxwell2D XY transient designs that use multi-slice skew models. Both single and two level distributions are supported.

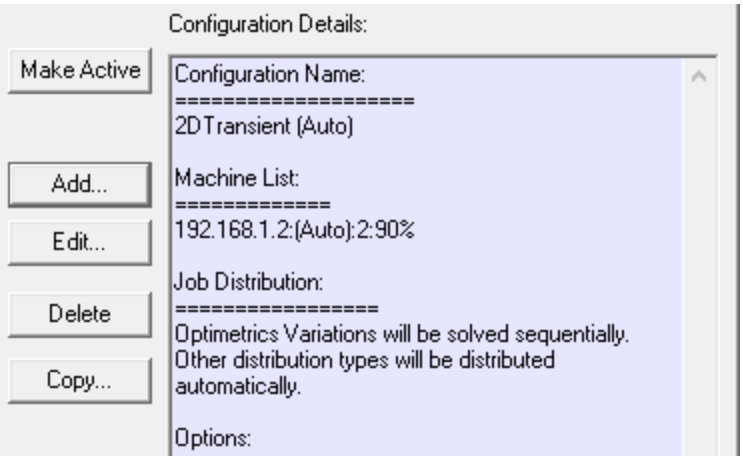
Note that the enabled distribution types will apply to all setups of the given design type, so it is possible for different setups in a design to be solved using different distribution types.

The distribution types that you enable here are listed in the HPC and Analysis Settings dialog box in the Configuration details pane when select that configuration from the list.





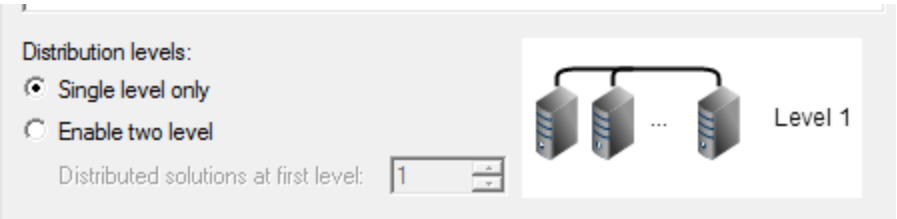
If your solver permits, and you have selected **Use Automatic Settings**, the Job Distribution notes that Optimetrics Variations will be solved sequentially. Other distribution types will be distributed automatically.



For products that support two-level distribution, when the design is appropriate, you can turn on two level distributed solves, and specify the number of distributed solutions for level 1.

Distribution Levels

The radio buttons let you specify **Single level only** or **Enable two level** distribution.



If you select Single level only, one distribution type will be applied at each stage of the solution process. If multiple types are available, the higher level solution will generally be distributed. All machine tasks will be used by the single-level distribution.

Single Level Distributions

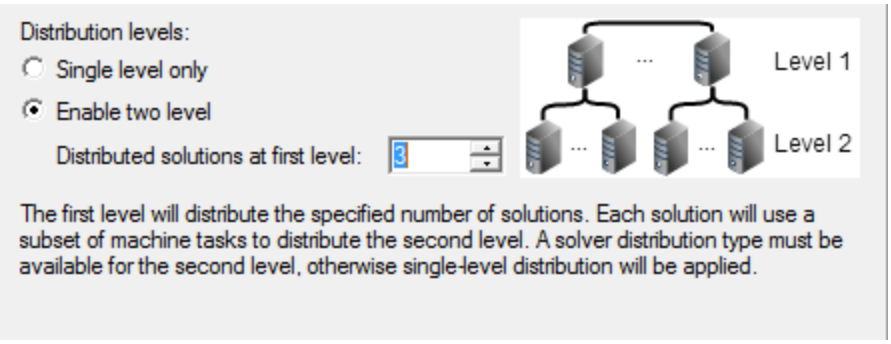
Supported distribution types are **Optimetrics Variations**, **Frequencies**, **Transient Solver**, **Solution Matrix**, and **Skew Model**. Frequencies and solver distributions require MPI.

- Parallel distribution types such as **Optimetrics Variations**, and **Frequencies** are considered as not required. If these types are not able to distribute, the simulation can be run sequentially.
- **Transient Solver**, **Solution Matrix**, and **Skew Model** distribution types are considered as required. If these types are enabled the software will assume that distribution is necessary to extend the simulation scale or add fundamental solution capabilities.
- When multiple distribution types are available, the higher level solution will generally be distributed. The priority order is : **Optimetrics Variations**, **Transient Solver**, **Frequencies**, **Solution Matrix**, and **Skew Model**. If Iterative Solver is enabled in the Eddy Current solve setup, the priority order will be: **Solution Matrix**, **Optimetrics Variations**,

Transient Solver, and **Frequencies**. For example, **Optimetrics Variations** will be distributed when both **Optimetrics Variations** and **Transient Solver** are selected.

Two Level Distributions

Selecting **Enable two level** enables the Distributed Solutions at first level selection box.



The first level will distribute the specified number of solutions. Each solution will use a subset of machine tasks to distribute the second level. The **Transient Solver** distribution type must be available for the second level for a Transient design; otherwise single-level distribution will be applied. The **Solution Matrix** or **Frequencies** distribution type must be available for the second level for an Eddy Current design; otherwise single-level distribution will be applied.

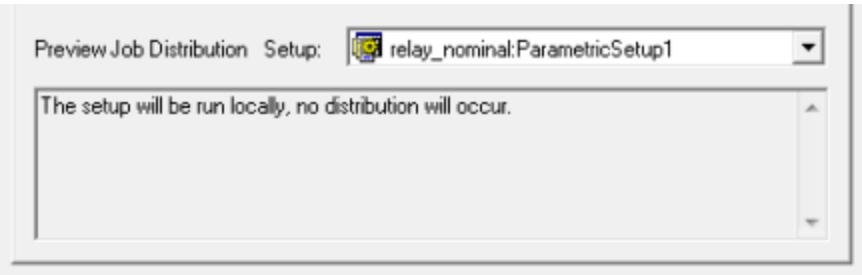
- Distribution types that can be distributed at Level 1 are **Optimetrics Variations**, and **Frequencies**(Maxwell 3D).
- **Frequencies**, **Transient Solver**, **Solution Matrix** (Maxwell 3D only), and **Skew Model** (Maxwell 2D XY transient only) distribution types can be distributed as Level 2.

The following are examples of a two-level distribution:

- A parametric setup distributing **Optimetrics Variations** as level 1, and a **Frequencies** distribution as level 2.
- A parametric setup distributing **Optimetrics Variations** as level 1, and a **Skew Model** distribution as level 2.

Preview Job Distribution

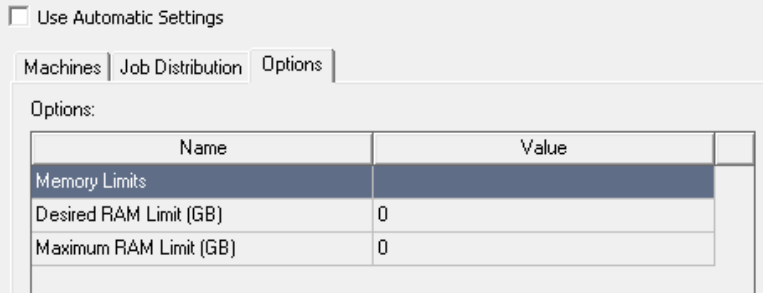
The Preview Job Distribution Setup menu and field lets you view how the selected setup will be distributed.



Options Tab (Analysis Configuration Dialog)

Use the **Options** tab to specify options for the current analysis configuration. Different design types may have different options available in their analysis configurations.

Note	The Options tab is not displayed for RMxprt analysis configurations.

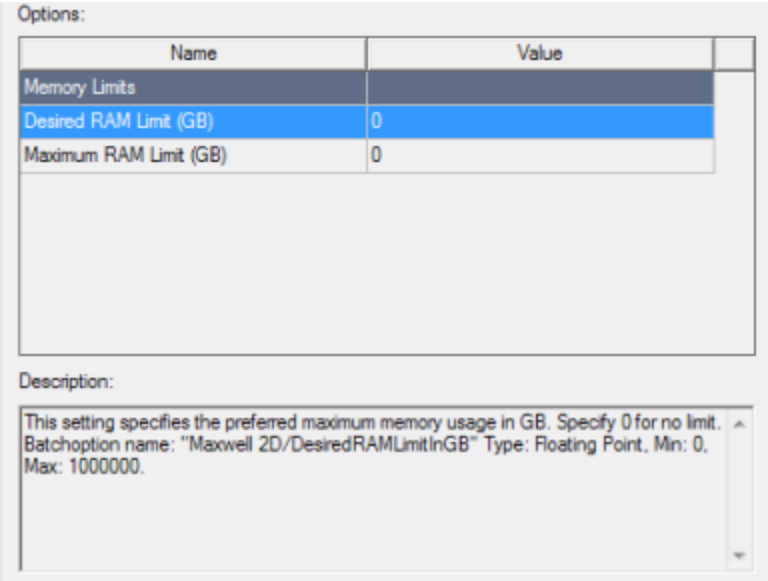


The Options grid contains the available option Names and editable Values.

These options settings will be in effect only when all the following are true:

- A design is being solved whose design type matches this analysis configuration's
- This analysis configuration is the active configuration for its design type
- You have not specified corresponding batch options on the command line. Command line batchoptions can be used to override the options specified by the active configuration.

Selecting an available option causes a display of a description of the option.



Relation to Batchoptions

Analysis configuration option settings can be overwritten by specifying the option name and value inside a -batchoption string. See -batchoptionhelp for a list of batchoption names and possible values. You can also view all available and frequently used batchoptions in the Job Management **Submit Job To:** dialog box, by clicking the **Add...** button under Analysis Options. This opens the **Add Batchoption** dialog box, which gives access to all batchoptions.

Adding Configurations or Accepting Edits

Click **OK** to accept the changes and close the **Analysis Configuration** dialog box. Only machines checked as **Enabled** appear on the distributed machines Configuration Details Machine list..

Regardless of the machine(s) on which the analysis is actually run, the number of processors and RAM Limit (%), the Desired and Maximum RAM Limits, and the default process priority settings are now read from the machine from which you launch the analysis. See Setting HPC and Analysis Options.

For more information, see distributed analysis.

Note	The option is only active if there are multiple rows listed in the parametric table, there are multiple frequency sweeps listed under a given analysis setup, and the number of distributed analysis machines is two or greater.

Related Topics

Configuring Distributed Analysis

Editing Distributed Machine Configurations

Selecting an Optimal Configuration for Distributed Analysis

Time Decomposition Method for Maxwell Transient Designs

Model Setting Tab

Selecting an Optimal Configuration for Distributed Analysis

With the distributed solve option (DSO) it is possible to assign more than one machine in the DSO setup panel to a single computer - assuming the computer has multiple cores. For example, a quad core computer can be listed 4 times in the DSO machine list and each CPU will run a separate simulation in the sweep. This raises the question of how to configure a distributed solution option (DSO) in combination with the multiprocessing option to take optimal advantage of the available hardware. The simplest answer is that for the overwhelming majority of situations the speed improvement will be greater if additional machines are added to a DSO than if the same CPU's are added as multiple processors. That simple answer assumes that there is enough memory for the DSO simulations.

For multiple DSO simulations on a single machine the total memory needed is the sum of the memory used by each simulation. For example, assume we are running discrete frequency sweep and each frequency point needs 3.5GB. Our computer has only 8GB of RAM but is a quad core system. To keep the computer from going heavily into swap, which is highly inefficient, we would only want to assign this computer twice to the list of machines in the DSO setup. To take advantage of the remaining 2 CPU's then it would make sense to set the number of distributed processors to 2. This would be the optimum for this setup. In addition one would need to be sure that the amount of disk space available is also sufficient to fit all the requested simulations - typically hard drive space is not the limiting factor.

Related Topics

Configuring Distributed Analysis

Editing Distributed Machine Configurations

Large Scale DSO for Parametric Analysis

Large Scale DSO for parametric analysis operates through a non-graphical batch application called desktopjob. You can run the desktopjob command-line to perform parametric analysis DSO. The command-line interface supported by this batch program is similar to the command-line used for regular DSO jobs.

Large Scale DSO is used for 'large scale parallel' jobs, which either fail or scale poorly as Regular DSO jobs. A Large Scale DSO job does not support the output of full parametric results, but produces "reduced" datasets corresponding to predefined Rectangular plots. The extracted columns of data are saved as .csv files. Typically, there is one .csv file per-trace, per-variation. These .csv outputs can be used directly in downstream applications (for example, Excel, or custom programs that parse .csv files). They can also be Imported as Dataset Solutions for post processing. Non-Rectangular plots of the design (such as statistical eye or digital plot) are not extracted). In order to produce a new output you must re-run the analysis.

The basic process involves:

1. Prepare the project for Large Scale DSO Analysis
2. Submit the Large Scale DSO Job through the Tools>Job Management menu, or via a command line.
3. Monitor the job's progress
4. Post process the results

For details, refer to the following sections:

- Prerequisites for Large Scale DSO
- Job Management Interface for Large Scale DSO
- Large Scale DSO Example
- Command Line Syntax
- Deployment/Configuration
- Results Database Organization
- Job outputs
- Job Monitoring
- Known Issues for Large Scale DSO
- Troubleshooting for Large Scale DSO

Large Scale Distributed Solve Operation could submit a parametric setup to be solved in multiple machines, each machine may launch multiple EM-Desktop processes to solve the assigned variations (Design Points). Variations are distributed to each task (EM-Desktop process) equally, regardless of the machine hardware and each variation's complexity. In practice, some tasks may finish earlier than others, in some extreme case some tasks may hours behind fastest task. DSO can redistribute tasks when a task finishes before other task. Variations are removed from slow tasks and reassigned to fast tasks. If you abort a task, they can be re-assigned to the running task, when the running task finish its original assignment. For more information, see Large Scale DSO theory.

Large Scale DSO offers two new batchoptions related to the redistribution ability.

LargeScaleDSO/VarRedistribution, where 0 disables redistribution (default), and 1 enables it.

LargeScaleDSO/RedistributionLimit, is a positive integer specifying the minimum estimated remaining time (in minutes) for variations to redistribute to another task. The default is 3.

Aborting a Large Scale DSO Simulation

To abort the whole Job, select the **Abort** button on the Job Monitor dialog.

To abort using the Job progress bar, click the button next to the Job progress bar, Click the **Abort** menu item in the popup.

To abort all tasks in a Node(host), click the button next to the node progress bar., and click the **Abort** menu item in the popup. Aborted Variations will be redistributed to other running Nodes, if redistribution is turned on.

To abort an individual task, click the button next to the node progress bar, and click the "Detail" menu item in the popup. In the Task status dialog box, click the **Task** button in the grid. In the variation status dialog box, click **Abort** button. Aborted Variations will be redistributed to other running tasks, if redistribution is turned on.

To terminate a hanging EM-Desktop process, the hanging ansysedt.exe process won't respond to the first abort command. Send a second abort command to terminate the hanging process. Remaining variations will be redistributed to other running tasks, if redistribution is turned on.

For an EM-Desktop process crashed or killed by Windows Task manager or other tools, the task status will be shown as aborted, Remaining variations will be redistributed to other running tasks, if redistribution is turned on.

Related Topics

Running Maxwell from a Command Line

Distributed Analysis

High Performance Computing (HPC) Integration

Large Scale DSO Theory

Prerequisites for Large Scale DSO

General Prerequisites

- Ansys Electronics Desktop must be installed on the cluster which runs either a supported scheduler or Ansys RSM.
- The cluster is compatible with large Scale DSO Requirements
- For releases before 19.0, the input projects for Large Scale DSO jobs distributed to multiple hosts are normally required to be on a shared drive. Beginning with the 19.0 release, this requirement is eliminated for most cases. For the RSM scenario, the shared drive requirement is eliminated, but the project must be accessible on the local host, which must be the first host for the job. For the scheduler scenario, because the job can get started on any of the compute nodes based upon what is allocated to the job by scheduler, users may still need to have the project on a shared drive. However some customers have their own job scripts for job spooling, which copy the project from the source location to a local folder on the first execution host of the job. Such customers will no longer need shared drive for their projects.
- Every node of the cluster supports the disk space (in temp directory) and memory requirements of multiple engines that run in parallel.
- All the machines allocated to Large Scale DSO job must all come from the same platform, Windows or LINUX.

RSM Environment

On the Windows platform, RSM (ansoftrsm.exe) is started as an 'admin' account, rather than as a 'system' account.

Note	Large Scale DSO does not support RSM Service running with 'system login' credentials.

- On each machine of the cluster, 'desktopjob' application is registered with RSM service using the command shown below:

  Windows: <installation-directory>/<platform> desktopjob.exe -regserver

Linux: <installation-directory>/<platform>desktopjob -regserver

Note	Detailed cluster configuration instructions are here.

Scheduler Environment

- No extra configuration is needed.

Related Topics

Large Scale DSO for Parametric Analysis

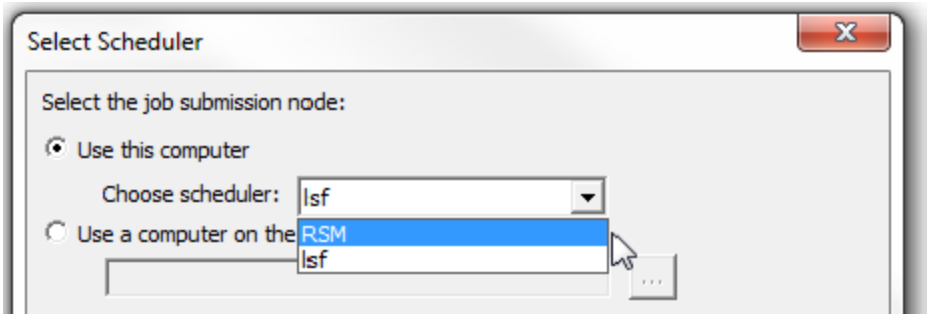
Distributed Analysis

High Performance Computing (HPC) Integration

Job Management Interface for Large Scale DSO

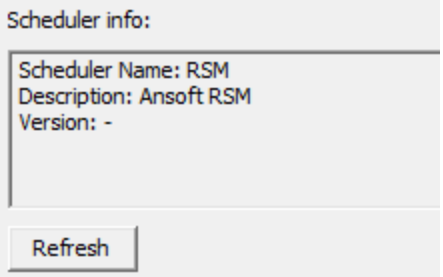
Large Scale DSO jobs run only in non-graphical batch mode, irrespective of the scheduler environment. This is in contrast to a Regular DSO job, which can run in graphical mode. This consideration implies that projects corresponding to Large Scale DSO job must be saved and closed prior to job submission. Secondly, the command to submit a Large Scale DSO job is only available through **Tools** > **Job Management** or via a command window, while a Regular DSO job can be run in an RSM environment by right-clicking directly on the parametric setup. The Job Management window is accessed by running Ansys Electromagnetics product Desktop on the designated Postprocessing Node of the cluster. The Desktop provides UI commands for Scheduler selection, Job submission and Job monitoring/control. You access the Scheduler User Interface by clicking **Tools>Job Management>Select Scheduler...**

Click Select Scheduler to display the selection dialog box. A drop down lists potential schedulers, (which can include RSM, lsf, or sge, depending on the environment).



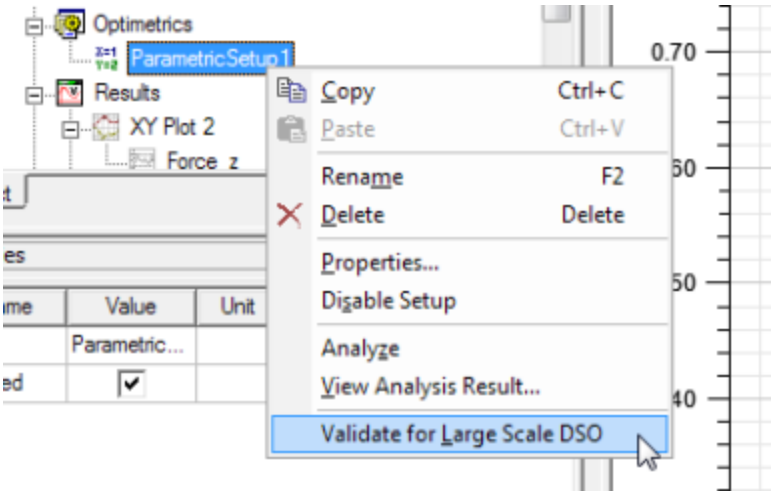
If you select a scheduler that is not supported in your environment, you receive a warning message.

After selecting a scheduler, you can click Refresh to display information for that scheduler.



When you have selected a scheduler supported in your environment, perform the following steps to submit a Large Scale DSO job.

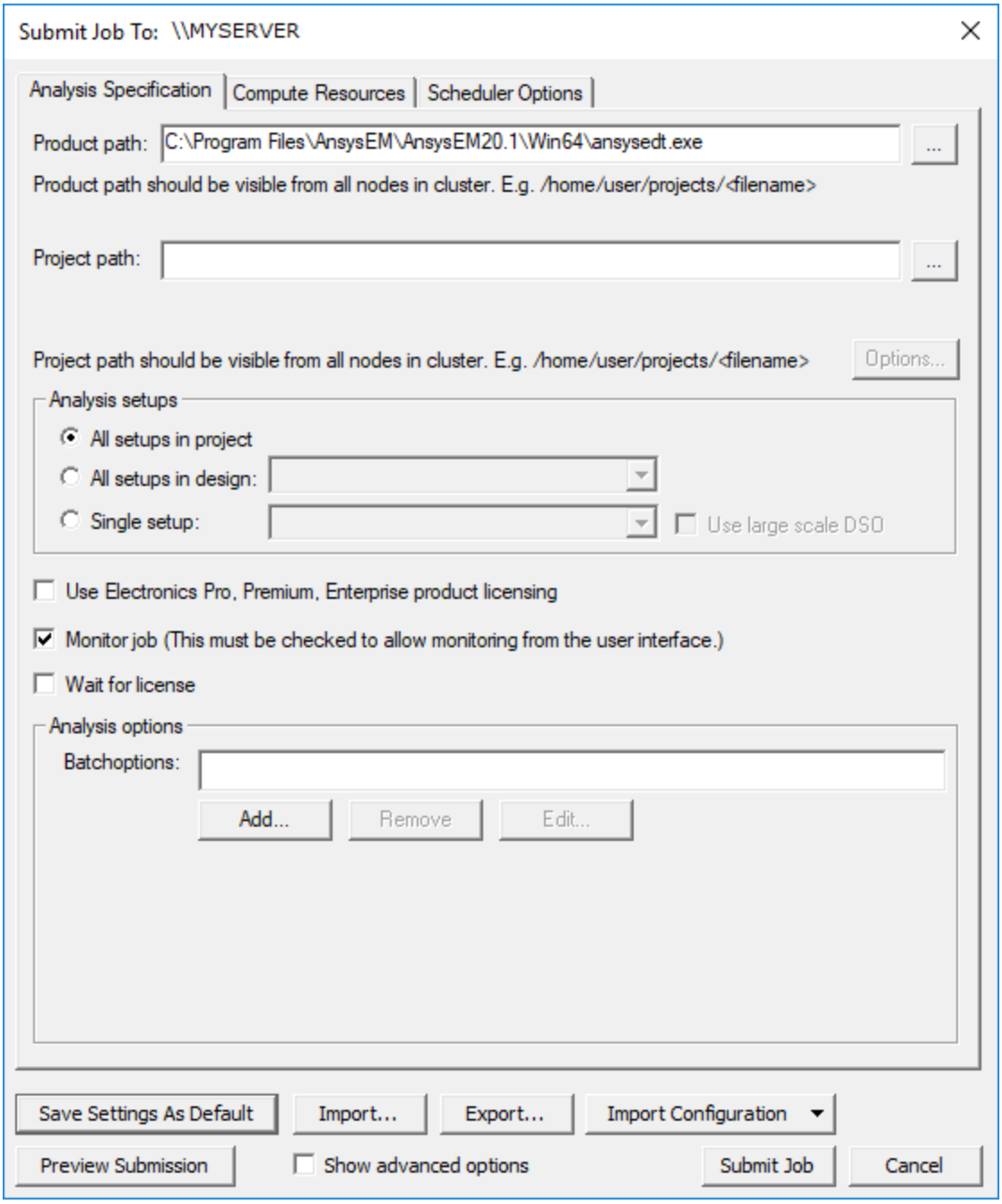
1. Set up and prepare the model on your local workstation. Right-click the desired solution setup and select **Validate for Large Scale DSO**.



2. Correct any errors and save the project.
3. Copy the project (or folder, if the project references external files) from a personal workstation to a shared drive on cluster.
   - In the RSM environment, you must specify a machine list (See Editing Distributed Machine Configurations).
   - In a Linux scheduler environment, a cluster must have a designated 'postprocessing node'.
4. Open a remote desktop session (or equivalent) on the node corresponding to the first machine in the machine list (the designated Postprocessing Node on Linux).
5. Launch Electronics Desktop on that node and open the project.
6. Verify that the model has been prepared correctly.

7. Close the project.
8. Submit the job one of several ways:
    - Click **Tools** > **Job Management** > **Submit Job**.
    - Click **Project** > **Submit Job**.
    - Click <*product_name*> > **Submit Job**.
    - Select the **Simulation** tab and click the **Submit** icon.

   The **Submit Job To** window appears, on the **Analysis Specification** tab.



9. Run **Tools>Job Management>Submit Job...** The standard Job Submission panel pops up.

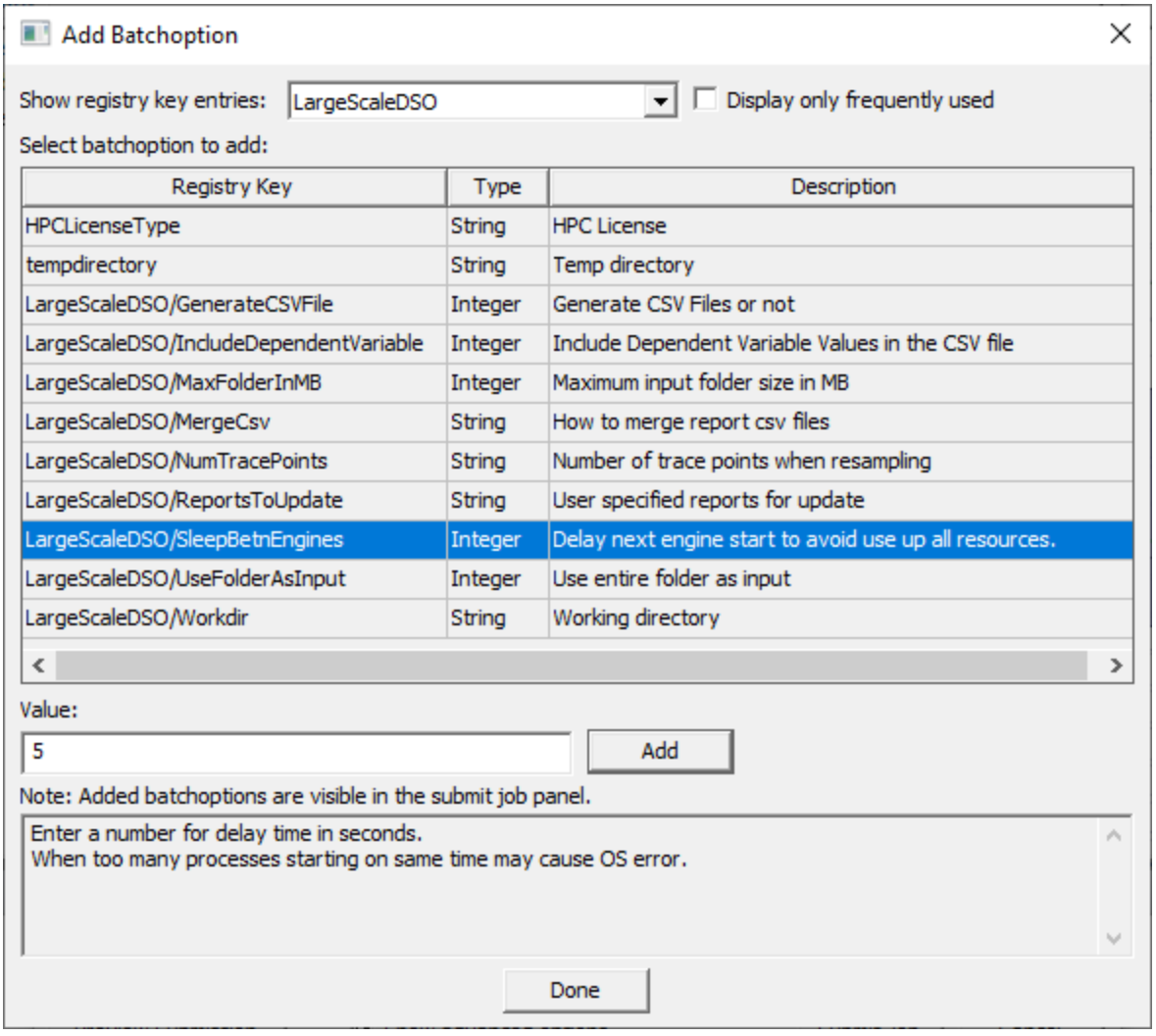
> **Note:** Options vary slightly depending on the selected scheduler.

10. Specify information for all fields:

- Most options will not be enabled until you select a **Project path**.
- Under **Analysis setups**, select setups for analysis.

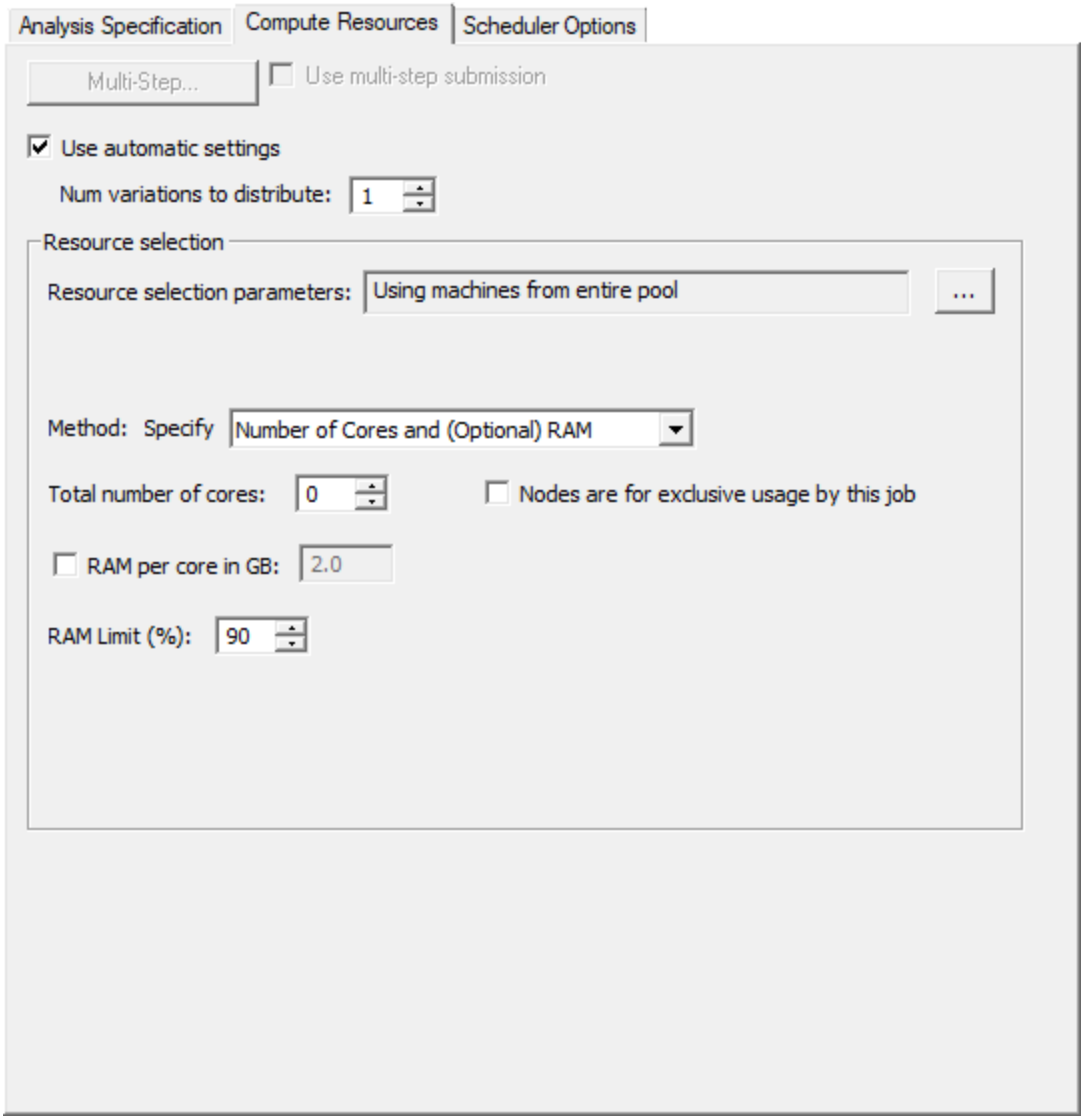
  For Large Scale DSO, select a **Single setup** from the drop-down menu and ensure the **Use large scale DSO** check box is selected.
- Determine whether to **Use Electronics Pro, Premium, Enterprise product licensing**, whether to **Monitor job**, and whether to **Wait for license.** If you wish to monitor jobs through **Tools** > **Job Management**, you must enable job monitoring.
- Add **Batchoptions**, if desired.

  The following shows batchoptions specific to Large Scale DSO.



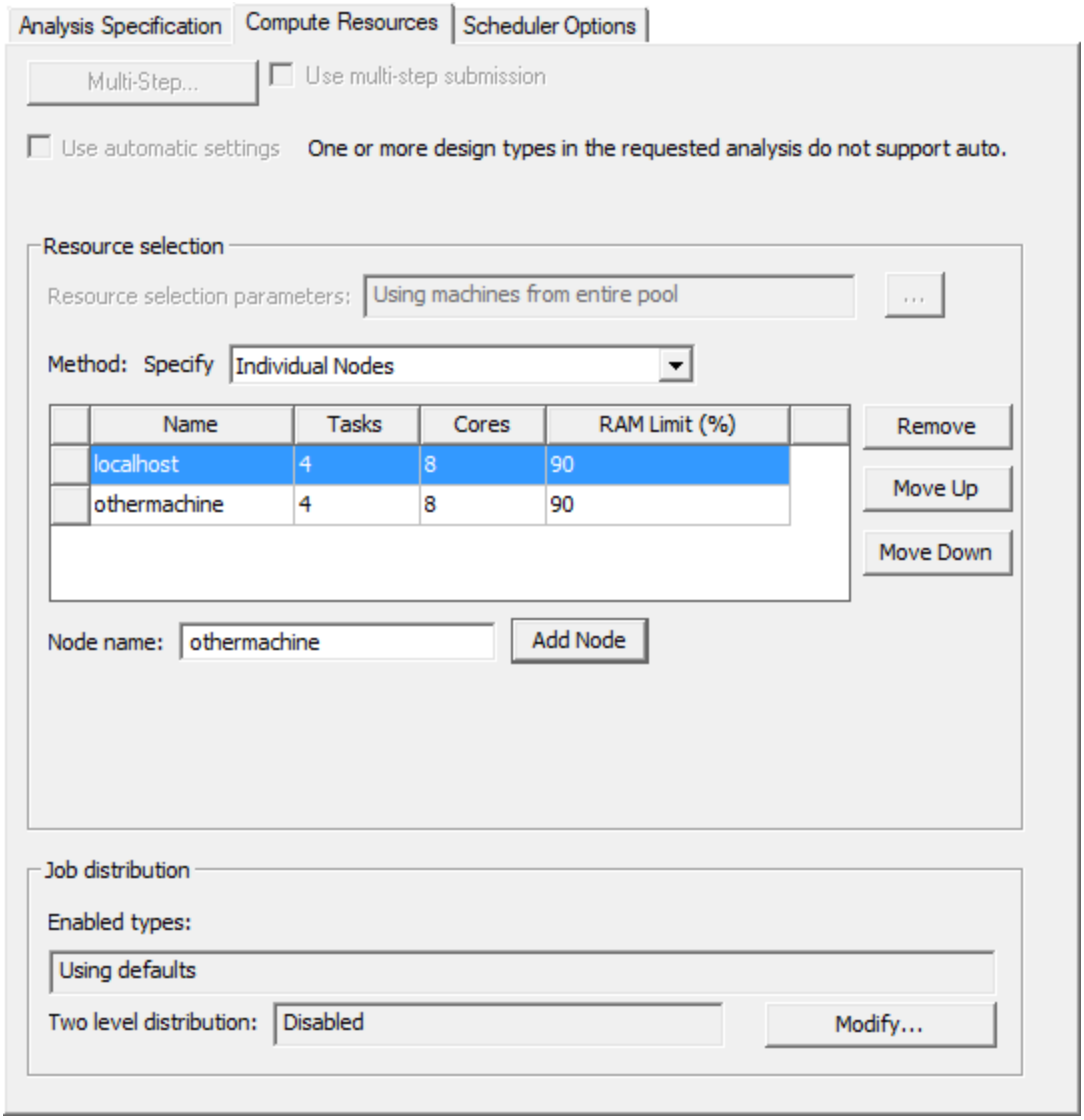
11. Select the **Compute Resources** tab.

    When available, **Use automatic settings** may be selected. If it is selected, also select the Number of variations to distribute.



The values you specify represent minimal requirements for each condition that can interact in leading to the total resources the Scheduler derives from them. A submission preview shows the number of resources assigned.

When automatic settings are not available or are not selected, additional options appear:



For RSM Large Scale DSO jobs submitted from the Job Submission panel, localhost must be the first node in the resource selection panel. Otherwise, the job will fail.

In the **Job distribution** area, you can enable or disable **Two-level distribution** by clicking **Modify**. See: Two-level Distribution Guidelines.

12. If desired, click **Preview Submission** to view a summary of the commands to be sent to the scheduler. The text can be copied to the clipboard.
13. To submit the commands to the scheduler, click **Submit Job**.

Note	The RSM environment does not support queuing, so clicking **Submit Job** starts it immediately.

14. If you enabled job monitoring, you can monitor the job via **Tools**> **Job Management** > **Monitor Jobs**. See: Monitoring Jobs.

Large Scale DSO Command Line Syntax

Large Scale DSO feature operates through a non-graphical batch application called 'desktopjob'. You can run the desktopjob command-line to perform parametric analysis DSO. The command-line interface supported by this batch program is consistent with the command-line used for current DSO jobs. “desktopjob -help” lists all available command-line options as shown below:

Command Line Syntax:

desktopjob.exe `<options> <project-path-on-shared-drive>`

Note that the project path can be to an archive file.

Options:

-help: Print this help text

-cmd: Specify command to run. Available choices: dso

-ng: Run analysis in non-graphical mode

-monitor: Output progress and messages to standard output/error

-waitforlicense: Queue the job until the availability of licenses

-useelectronicsppe: Use Electronics Pro Premium Enterprise licensing to override the Tools/Options entry. For example, -useelectronicsppe 1 to use that license, or -useelectronicsppe 0 to not use that license.

-preserve: Preserves the local storage space of the distributed job for investigation into the job's run. If local storage directory (for example, the temp directory) is provisioned by scheduler, ensure it is also configured to preserve the job's local storage. *This storage should be deleted manually*.

-batchoptions: Override the Tools/Option entries through either a batchoptions file or batchoptions string.

Example:

```
-batchoptions <config-file-on-shared-drive>
-batchoptions "'name1'='val1' 'n2'='v2'"
```

-jobid: Specify a custom job ID for the job. The job's output is organized into a folder with job ID name. This parameter is ignored when run under a scheduler.

-machinelist:

- In the context of **RSM**:

  Specify machines for distributed analysis. Machine list is specified either inline (as a comma separated machine names) or through a file. Multiple cores are specified by repeating the name of machine or by embedding number of cores in the machine name, using a colon separator.

  Example 1:

  ```
  -machinelist "list=m1,m1,m1,m2,m2,m3"
  ```

  Example 2:

```
-machinelist "list=m1:3,m2:2,m3"
```

Example 2:

```
-machinelist "file=machines.txt"
```

- In the context of **a scheduler such as LSF**:

  Specify the portion of total machines for distributed analysis. Use remaining for overhead or shared memory multiprocessing.

  Example:

  ```
  -machinelist "Num=10"
  ```

-auto: Run the leaf jobs in auto mode.

-usefolderasinput: Choose this option if the job's input represents the entire folder rather than just the project file.

-maxfolderInMB: Specify the maximum size (MBytes) of input folder that is allowed for a valid job. By default, the maximumsize allowed for input is 10MB. Specify a value of 0 to remove this size restriction and enable inputs of any size. Note: this option applies when '-usefolderasinput' is used.

-workdir: Specify the shared drive folder for status and result files generated by analysis. By default, the results folder of input project is used as the work directory.

-meregecsv: [acrossDPs | singleDP | both]:

across DPs: Merge report csv files for all design-points (variations). One file is created per trace, across all variations.
singleDP: Merge csv files within a single design-point (variation). One file is created per variation, per a set of traces that can be merged.
both: Merge all traces that have the same primary sweep for all design-points (variations) into one csv file.
Interpolation note: If primary sweep values are not uniformly spaced, mergecsv is enabled with traed values are re-sampled uniformly using '-batchoptions' syntax as shown below:

```
-batchoptions "'LargeScaleDSO/NumTracePoints'=500"
-batchoptions "'LargeScaleDSO/NumTracePoints'='PrimarySweepName:200'""
-batchoptions
"'LargeScaleDSO/NumTracePoints'='ReportName1:Trace1:100;ReportName1:Tr
ace1:100;ReportName1:TraceName2:200'"
```

-abort: Abort a running job identified through the job's working directory. Example: `-abort <projectresultsfolder-path>/>jobid>`. For a complete discussion of methods for aborting jobs or specific tasks, see the discussion of Aborting a Large Scale DSO Simulation under Large Scale DSO for Parametric Analysis.

-repackageresults: Choose this option to add simulations results to the input archive file. Note: this option only applies if an archive file is provided as input.

-batchsolve: Solve the specified parametric setup. Syntax for the setup:

```
<design-name>:Optimetrics:<parametric-setup>
```

Related Topics

Large Scale DSO for Parametric Analysis

Running Maxwell from a Command Line

Large Scale DSO Job outputs

A large-scale-dso analysis does not support the output of full parametric results. Instead, it extracts 'subset' results using predefined Rectangular plots, which are created by user before job is run. The extracted columns of data are saved as CSV files. Typically, there is one CSV file per-trace, per-variation. (Note: Non-Rectangular plots of the design (for e.g. statistical eye, digital plot) are not extracted). The outputs can be either imported as datasets for post-processing in the desktop also as function of parametric variations, or used directly in downstream applications (for example, Excel, or custom programs that parse .csv files).

CSV File contents

The initial header rows of CSV file define the solved variation. For each such row, the first column has variable name and the second column has variable's value. The row following variation rows has the name of primary sweep and the name(s) of extracted quantities. Subsequent rows contain 'data' - quantity values as a function of primary sweep. Below examples provide a context for the CSV file contents:

- Traces of S-parameter report: The data portion of CSV file has 2 columns of data: first column has Freq values and the second column has values for trace's s-parameter component
- Trace of a far field report: Suppose there is a farfield report with a trace ('magrE'), whose primary sweep is 'phi' and secondary sweep is 'theta'. Further suppose that two values of 'theta' are chosen and 'all' values of 'phi' are chosen. For this trace, the data portion of CSV file contains three columns of data: the first column has phi values, the second column has magrE values for the first value of theta, the third column has magrE values for second value of theta. The magrE output columns are titled as 'magrE_crv1', 'magrE_crv2' respectively
- (Advanced) 'Special' sweeps: In the case of a trace with 'special' primary sweep (such as the trace of a time domain quantity), one CSV file is created per curve of trace, per variation. These CSV files always have two columns, irrespective of the number of values chosen for secondary/higher sweeps.

Related Topics

Large Scale DSO for Parametric Analysis

Import Large Scale DSO Dataset Solution

Large Scale DSO Example

This section provides an example of the use of Large Scale DSO to distribute parametric variations of a Maxwell model across the nodes of a cluster or to multiple cores of a single machine.

This example presumes that your configuration satisfies the Prerequisites for Large Scale DSO. For this example, pre-suppose that we have a Windows cluster. Further suppose that the shared drive folder, which contains the input projects and computed results is at the location

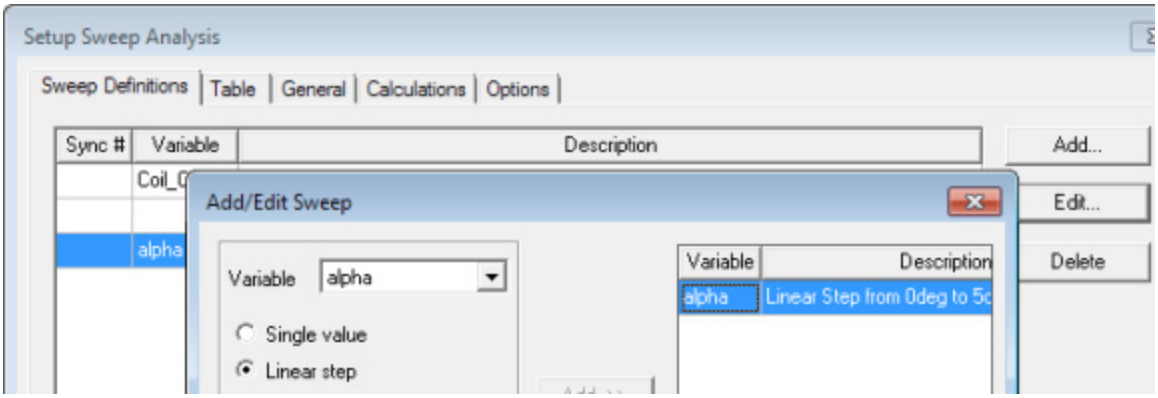
\\sjo7na1\mxwlprojs

Major steps for Large Scale DSO Example

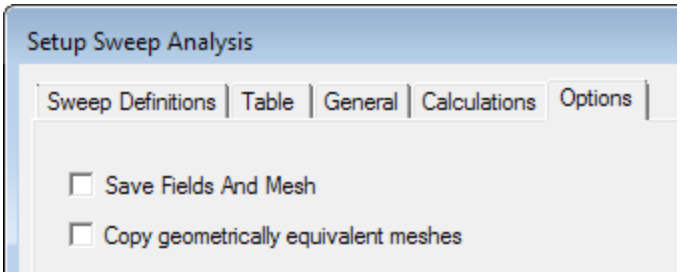
1. Prepare the model for Large Scale DSO Analysis
2. Submit the Large Scale DSO Job
3. Post process the results

Preparing a Project for Large Scale DSO Analysis

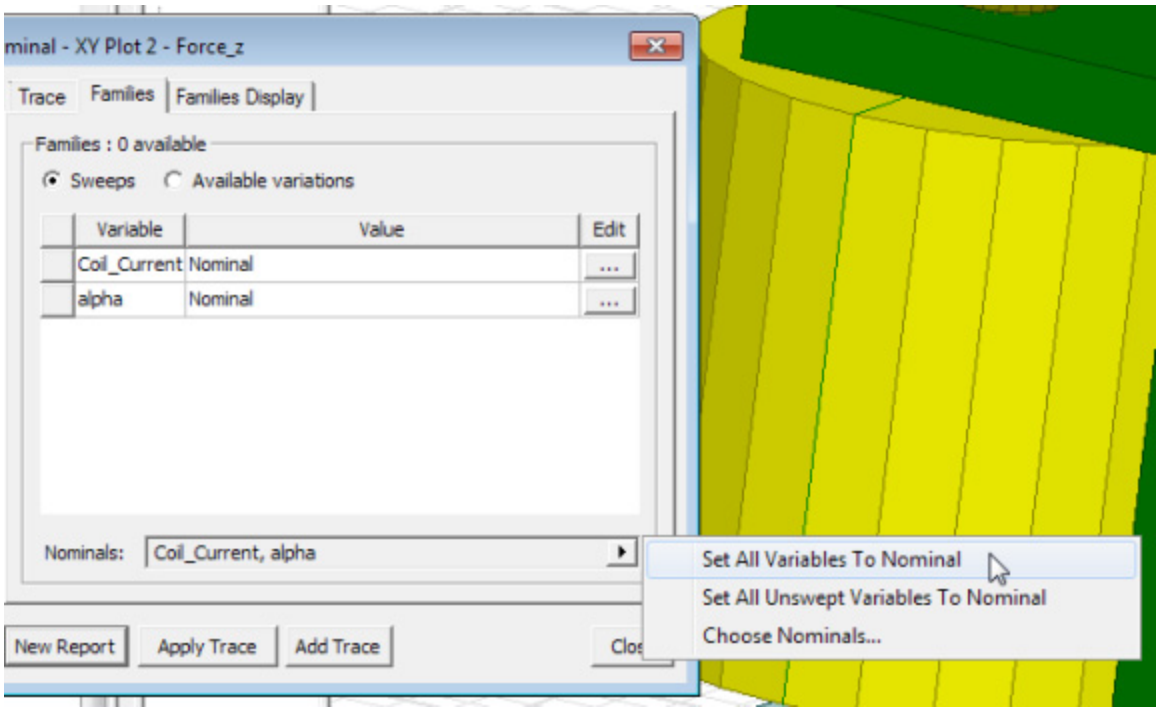
1. Ensure that the Maxwell project is on the shared drive. For this example, the **relay_nominal.aedt** file is on "\\sjo7na1\mxwlprojs\relay_nominal.aedt"
2. Set up a parametric table you wish to use.



3. On the **Options** tab, ensure that **Save Fields and Mesh** is unchecked.



4. Outputs from Large Scale DSO come from predefined rectangular plots that are created *before* the Analysis command is issued. Non-rectangular plots of the design (digital plots, for example) are not extracted. Follow the steps below:
   a. Because these DSO outputs come solely from Rectangular Plots, delete all other postprocessing setups.
   b. Use the **Report** window to define outputs for the desired rectangular plots.
   c. Click on the **Families** tab and ensure that all variables are set to **Nominal**, as shown below.



Note	For Large Scale DSO, outputs are not extracted correctly unless all variables on the **Families** tab are set to nominal.

   d. Click the **New Report** button to create the report.
5. Right-click on the parametric sweep, and select **Validate for Large Scale DSO**.

   A dialog reports any errors. The following are some typical messages:

```
[error] Please remove all the calculations in Parametric
Setup/Calculation page.
[error] Please turn off "Save Fields And Mesh" in Parametric
Setup/Options page.
[error] No rectangular plot exist in the design. Please create
a rectangular report.
```

6. Correct any errors, if necessary to pass validation.
7. Save and Close the project.

Submit the Large Scale DSO Job: Examples

This section includes examples of submitting a Large Scale DSO job using RSM and a Scheduler.

> **Note:** Functionality featured in the example(s) in this section applies to multiple design types.

Using the RSM Environment

1. Before submitting the job make sure that the input project is not open in any Desktop window.
2. For this example, suppose that there are two quad-core machines on your cluster with the names "m1" and "m2". Further suppose that there are two engines per machine, for a total of four parallel engines. Let the number of processors allocated to each engine be 1.
3. From a command prompt, issue the following command:

```
<installation-
directory>\v<version>\<platform>\desktopjob.exe -cmd dso -
machinelist "list=m1:2,m2:2" -batchoptions
\\sjo7na\mxwlprojs\mxwloptions.txt -batchsolve "relay_
nominal"Optimetrics:ParametricSetup1"
\\sjo7na\mxwlprojs\relay_nominal.aedt
```

   where the file \\sjo7na\mxwlproj\mxwloptions.txt has the following contents:

```
   $begin Config
'Desktop/Settings/ProjectOptions/NumberOfProcessors'=1
'Maxwell3D/Preferences/SaveBeforeSolving'=0
'Maxwell/Preferences/HPCLicenceType'='pack'
   #end 'Config'
```

4. Assume the above job is assigned ID "jobID".

Using a Scheduler Environment (such as LSF)

1. Suppose you want to solve variations using four parallel engines, each engine being assigned a single core.

2. From a command prompt, run the following command:

```
bsub -n 4 <installation-
directory>\v<version>\<platform>\desktopjob.exe -cmd dso -
batchoptions \\sjo7na\mxwlprojs\mxwloptions.txt -batchsolve
 "relay_
nominal:Optimetrics:ParametricSetup1"\\sjo7na\mxwlprojs\re
lay_nominal.aedt
```

   where the file \\sjo7na\mxwlproj\mxwlptions.txt has the same contents as the RSM example above.

3. Assume the above job is assigned an ID "jobid"

For example with 2 nodes of 44 cores each (with multiple distributed variations per machine):

MANUAL: `bsub -n 88 <installation-directory>\v<version>\<platform>\desktopjob -cmd dso -machinelist num=8 -monitor -useelectronicsppe=1 -ng -batchoptions " 'Maxwell 3D/NumCoresPerDistributedTask'=11 'Maxwell 3D/RAMLimitPercent'=90 'HPCLicenseType'='Pack'" -batchsolve TeeModel:Optimetrics:ParametricSetup1 OptimTee-v231-1.aedt`

AUTO: `bsub -n 88 <installation-directory>\v<version>\<platform>\desktopjob -cmd dso -machinelist numcores=88 -auto -NumDistributedVariations 8 -monitor -useelectronicsppe=1 -ng -batchoptions " 'Maxwell 3D/RAMLimitPercent'=90 'HPCLicenseType'='Pack'" -batchsolve TeeModel:Optimetrics:ParametricSetup1 OptimTee-v231-1.aedt`

For example, with 16 nodes of 44 cores each (note each distributed variation will span 2 machines):

AUTO: `bsub -n 704 <installation-directory>\v<version>\<platform>\desktopjob -cmd dso -machinelist numcores=704 -auto -NumDistributedVariations 8 -monitor -useelectronicsppe=1 -ng -batchoptions " 'Maxwell 3D/RAMLimitPercent'=90 'HPCLicenseType'='Pack'" -batchsolve TeeModel:Optimetrics:ParametricSetup1 OptimTee-v231-1.aedt`

Next

Large Scale DSO Example: Post Process the Results

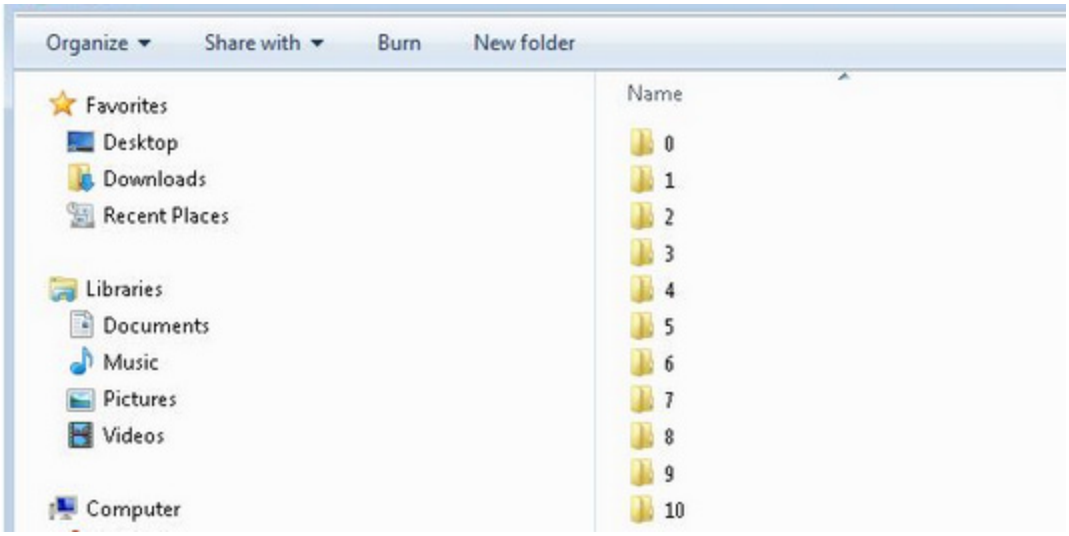
Related Topics

Large Scale DSO for Parametric Analysis

Large Scale DSO Example: Post Process the Results

Once the job is done, output is available in the ~\relay_nominal.aedt\jobid\results folder. Each variation creates a subfolder, which in turn has one csv file per trace of each report. See the detailed information regarding job monitoring and the location of the analysis logs.

The following figure shows the results for 10 variations as located in 10 folders.



Within each folder will be csv files corresponding to the traces for each variation.

You have three options for post-processing csv files.

- Import Large Scale DSO Dataset Solution
- Use Microsoft Excel or any other application that has csv post processing functionality.
- Parse the csv output into your custom program, for any downstream flow.

Related Topics

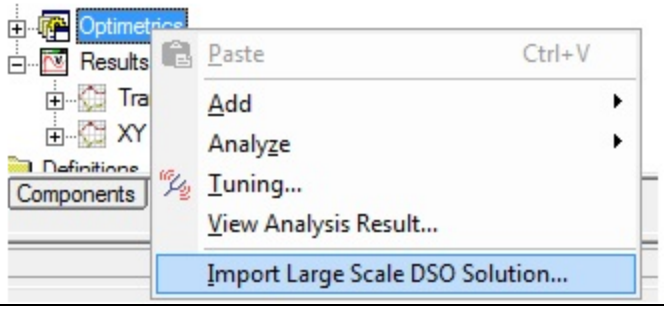
Large Scale DSO for Parametric Analysis

Import Large Scale DSO Dataset Solution

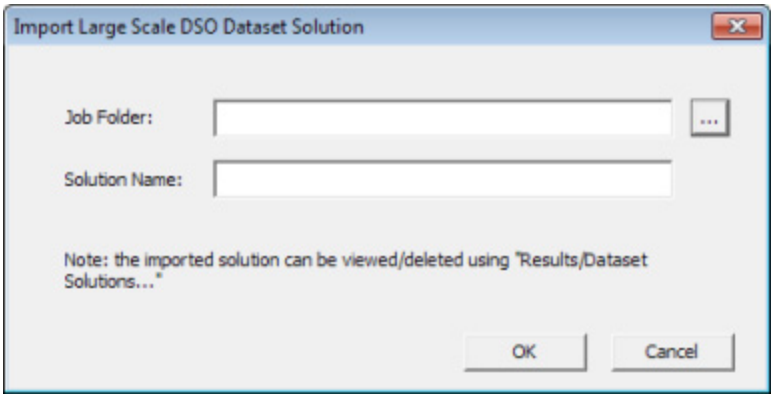
For post-processing Large Scale DSO dataset solutions in the desktop, you create a dataset solution through the **Import Large Scale DSO Solution** command and pointing to Large Scale DSO job's top level results folder.

1. Import solved large scale DSO solution. You can do this in two ways.

   In the Project tree. right click on Optimetrics and from the menu click **Import Large Scale DSO Solution**.



This opens the **Import Large Scale DSO Dataset Solution** dialog.

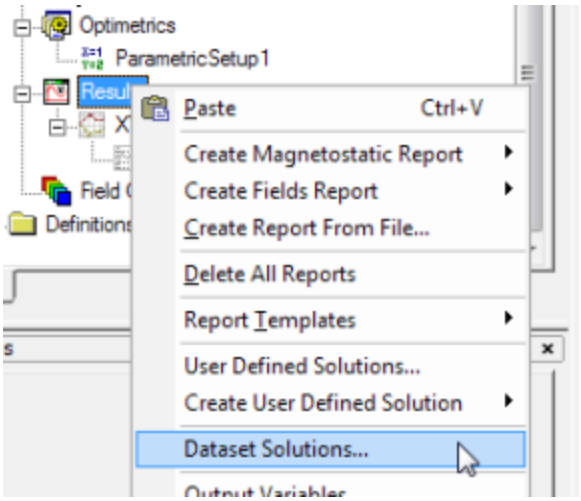


From here you can browse to the results of a Large Scale DSO job and select a job folder. Click the ellipsis button [...] to open the browser window. Navigate to the results directory to see the results, organized by the scheduler prefix and job ID number. To select a results dataset, double-click on the results folder name. See Large Scale DSO Results Database Organization.

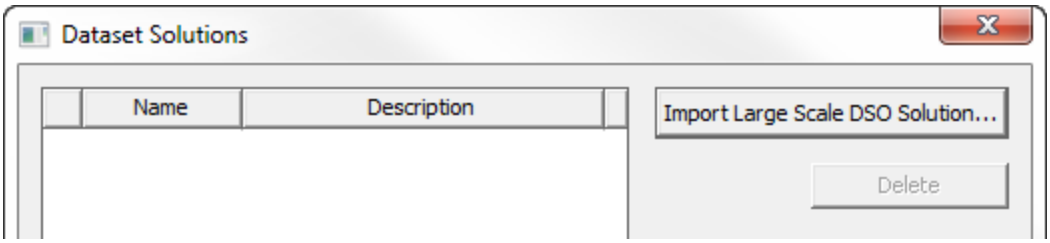
Click, **OK** to import the Dataset. If you have selected the solution folder correctly the dialog closes. If not, an error dialog opens.

2. To View/Delete created dataset solutions:

   In the Project tree, right click on Results and from the menu click **Dataset Solutions....**



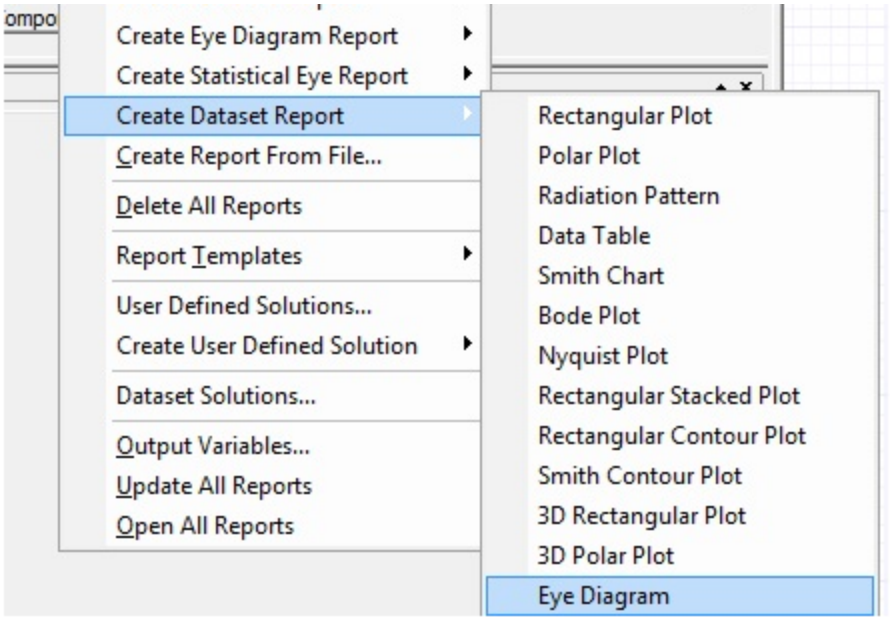
This displays the Dataset Solutions dialog.



The dialog lists any existing datasets. You can delete one or more by selecting from the list and selecting **Delete**. The **ImportLarge Scale DSO Solution** button opens the dialog for importing a dataset solution.

3. To create a Dataset Report

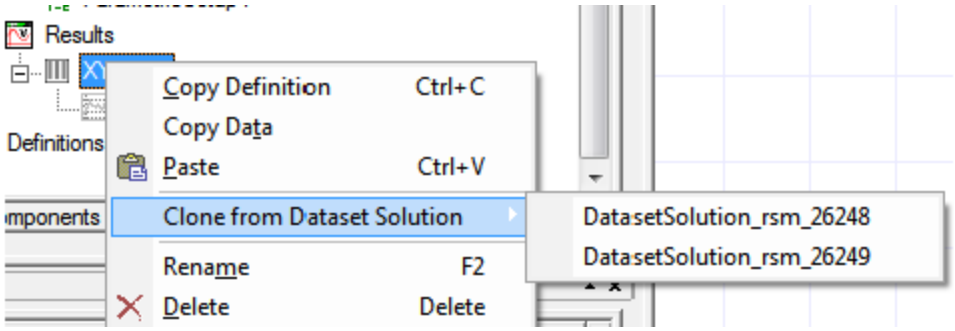
   After you have imported one or more DSO solutions, you can create a Dataset Report. Right click on Results and from the menu select **Create Dataset Report**, and the type of report.



   This opens a reporter window from which you can create a report. If you have previously created an eye diagram report and it is included in the DSO solution data extraction, you can use **Create Dataset Report > Eye Diagram** to recreate this report.

4. Cloning from a Dataset Solution

   If you re-open a project that was solved using large scale DSO, you can quickly clone a report for a solved large scale DSO solution by right-clicking in the Project tree on the report, and choosing Clone from Dataset Solution..., provided this report is qualified for Large Scale DSO data extraction. The provides a way that you can reuse the existing report definition and save the work of creating new report.

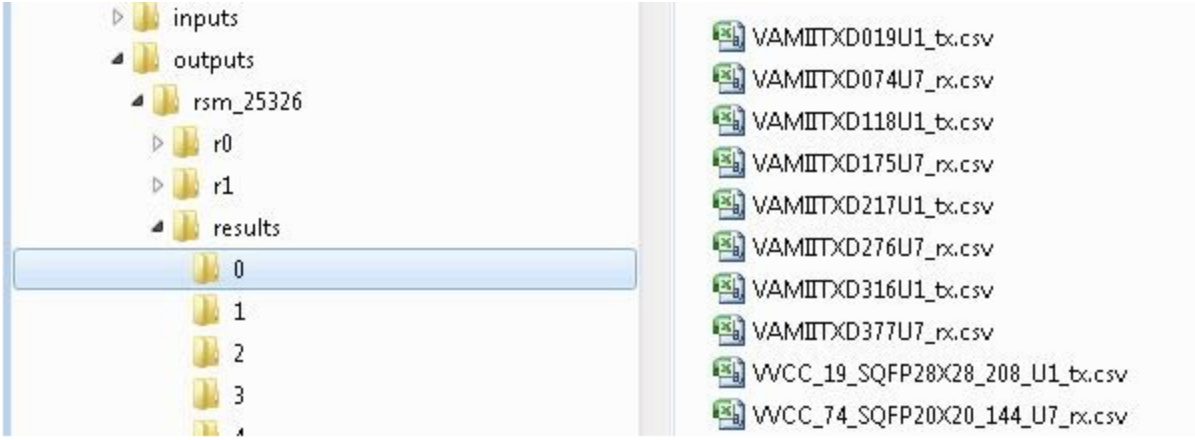


Large Scale DSO Results Database Organization

A Large Scale DSO analysis does not support the output of full parametric results. Instead, it extracts 'subset' results using predefined Rectangular plots, which are created by the user before the job is run. The extracted columns of data are saved as CSV files. Typically, there is one CSV file per-trace, per-variation. (Note: Non-Rectangular plots of the design (for e.g. statistical eye, digital plot) are not extracted)

The results of a Large Scale DSO job are located in the '<workdir>/<jobid>/results' folder. If 'workdir' is not specified on the job command-line, it is same as the input project's results folder. For example the default 'workdir' corresponding to '\\shared\projects\relay.aedt' is '\\shared\projects\relay.aedtresults'. Within this results folder, there is one folder per variation. The name of the variation's folder is an integer number corresponding to variation's index in the parametric table. For example, a variation-folder named '4' has results for the fifth row of parametric table, while a variation-folder named '0' has results for the first row of the table.

Below is a sample results folder showing the contents corresponding to results of first variation. There are ten CSV files corresponding to ten predefined traces.



Related Topics

Large Scale DSO for Parametric Analysis

Large Scale DSO Job Monitoring

Large Scale DSO avoids detailed intra-variation monitoring is avoided as it increases network traffic for large-scale jobs. Large Scale DSO jobs are monitored as below:

- **Cluster Monitoring Tools**: The resource usage (CPU, Memory, Network) of Large Scale DSO jobs is monitored using standard cluster monitoring tools. Such job-neutral resource monitoring is ideal as it uses negligible network bandwidth, CPU/Memory.
- **Detailed Monitoring of Analysis of a Variation**: For any detailed monitoring you must examine the information provided in the job's log files. Specifically, the large-scale-dso job writes detailed logs conveying information regarding the machines where engines are running and the local storage location of per-engine distributed database. With such information, you can login to individual machines for deeper probing of each distributed engine.

The following logs are available:

- **Per-node Logs**:

  There is one 'desktopjob.log' file per node assigned to the job. This log contains information regarding the node such as name, local storage folder, number of engines started on this node, etc. It is located in <workdir>/<jobid>/r<nodeIndex>. E.g. "<workdir>/<jobid>/r0" has desktopjob.log corresponding to the engines running on the first node of job, while "<workdir>/<jobid>/r2" has logs corresponding to engines running on third node
- **Per-engine Logs**:

  There is one desktopjob.log file per distributed engine. It is located in <workdir>/<jobid>/r<nodeIndex>/r<taskIndex>. For example, "<workdir>/<jobid>/r0/r0" has logs corresponding to first engine running on first node, while "<workdir>/<jobid>/r1/r2" has logs corresponding to third engine running on second node. Engine unique information (such as local storage of this engine) is logged here
- **Parametric Analysis Log**:

  This log file is located in '<workdir>/<jobid>/r<nodeIndex>/r<taskIndex>' folder and corresponds to Desktop's local-machine parametric 'batchsolve'. It is available only at the end of analysis and contains information regarding the variations solved by this engine and any info/warning/error messages.
- **Root Log**:

  This is the top-level log that logs job distribution information such as hierarchical activation and the list of nodes assigned to this job

For a complete discussion of methods for aborting jobs or specific tasks, see the discussion of Aborting a Large Scale DSO Simulation under Large Scale DSO for Parametric Analysis.

Related Topics

Large Scale DSO for Parametric Analysis

Large Scale DSO Deployment/Configuration

Linux Cluster configuration

- Shared drive for projects: Cluster must provide a shared drive that hosts job inputs - the submitted project must be located on a shared drive (for example, a sub-folder of user's home directory). The shared-drive must accessible using the same path on every node of cluster
- 'Temp directory' configuration

  Temp directory is either on 'local storage' or on storage that has equivalent speed characteristics i.e. the I/O rates of the storage should be invariant to network traffic

  Temp directory on a host has sufficient space to hold results database for the variations that are solved on it. Note:

  This storage is freed at the end of the analysis

  The amount of required space depends on the number of engines per node and the cumulative variations solved on this node

  The amount of required space depends on the project's compression-options. For e.g. if 'Save Fields' of a parametric setup is OFF, the space requirement is smaller by the amount of space taken up by field solution data
- Ansys Electromagnetics RSM environment: In the case of supported scheduler environments, there is no extra configuration needed. In the case of RSM environment, following additional steps are needed:

  RSM must be running on all the nodes of cluster. The credentials of 'RSM service' allow read/write to shared drive. Reason: the remote engine processes are launched using the credentials of RSM service

  Registration of 'desktopjob.exe' with RSM service: 'desktopjob' program must be registered with Ansys Electromagnetics RSM using 'desktopjob -regserver". To ensure that the registration is successful, check that the 'desktopjob' entry in '<RSM-installation-folder>/AnsoftRSMService.cfg' file is valid.

Note	Linux-specific critical note: Edit AnsoftRSMService.cfg and replace 'desktopjob.bin' with 'desktopjob'

Major limitation: In the RSM environment, Large Scale DSO can only be enabled for one product.

Troubleshooting hints (RSM environment only): "shared drive read/write" requirement is a new constraint introduced in Large Scale DSO. So if a user runs into a situation where Regular DSO jobs run and Large Scale DSO jobs fail, one possible cause for the failure: RSM service does not have privileges to read and write to project folder located on shared-drive.

Windows Cluster configuration

All the above steps apply, except for steps that are stated as Linux-specific. Additional instructions:

- RSM and Ansys Electromagnetics products are either installed locally on each node of cluster (i.e. local installation) OR installed on a single shared-drive available to all nodes of

cluster (i.e. network installation)

- Registration of 'desktopjob.exe' with RSM service:
- Network installation: desktopjob.exe is registered with RSM service once, on any of the nodes of cluster
- Local installation: Since each node has it's own RSM installation, desktopjob.exe must be registered with RSM on each node.

Note	IMPORTANT! RSM service must be started using the credentials of a non-system 'admin' account, which has read/write permissions to project's shared drive. If RSM service runs as 'system' user, large-scale-dso jobs will fail

Heterogeneous Cluster configuration

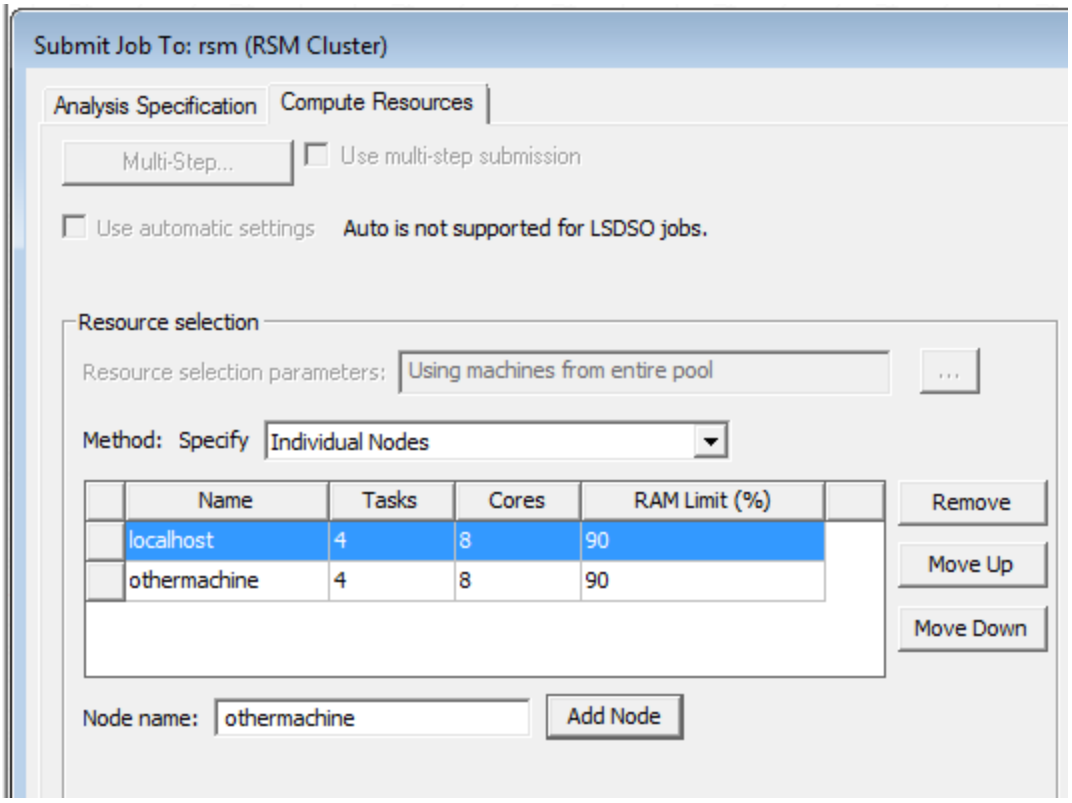
Limitation: Currently heterogeneous cluster (with both linux and windows nodes) is not supported. This is due to the shared drive requirement.

Related Topics

Large Scale DSO for Parametric Analysis

Known Issues for Large Scale DSO

For Large Scale DSO jobs that are submitted from job submission panel using RSM, localhost must be the first node in the resource selection node list, other wise Large Scale DSO solve with RSM will fail.



Cluster configuration (shared drive requirement): The input files (project, etc.) must be present on a shared drive that is accessible from every node of the cluster.

Note: this item is actually listed as a requirement.

Related Topics

Large Scale DSO for Parametric Analysis

Troubleshooting for Large Scale DSO

(Linux only) Deployment/Installation errors (such as mainsoft related or related to deployment configuration) are not captured. If there is such an issue, Large Scale DSO job will fail without useful messages in the logs

Report-based extraction fails mysteriously if traces and parametric-setup are not 'prepared' as per the Getting Started guides.

Job monitoring and control

Job re-start: There is no provision for stopping and re-starting a job. A new job does not reuse solved results - always solves all rows in the table. So an abort or failure of a job re-starts from the

beginning, unless a new parametric table with the unsolved rows is created

Job outcome

Job status: The exit code of job doesn't indicate success or failure correctly. The error messages from multiple log files needs to be combined to determine the reason for failure. In many situations, the reason for a failure is apparent only after re-running the job after turning ON the 'debug logging'

In some Linux scenarios, the analysis appears to finish successfully with valid results, except that the exit code is '134'. In this case, although the exit is abnormal, the failed exit code can be ignored

Load Balancing: For models with 'unbalanced variations table' (i.e. variations that take considerably different amount of time to solve are clustered in few regions of table), job will take longer time to solve than a Regular DSO as the job's overall completion time is determined by the slowest solving region.

Workaround: rearrange the rows in the parametric table so that each region takes a similar time to solve

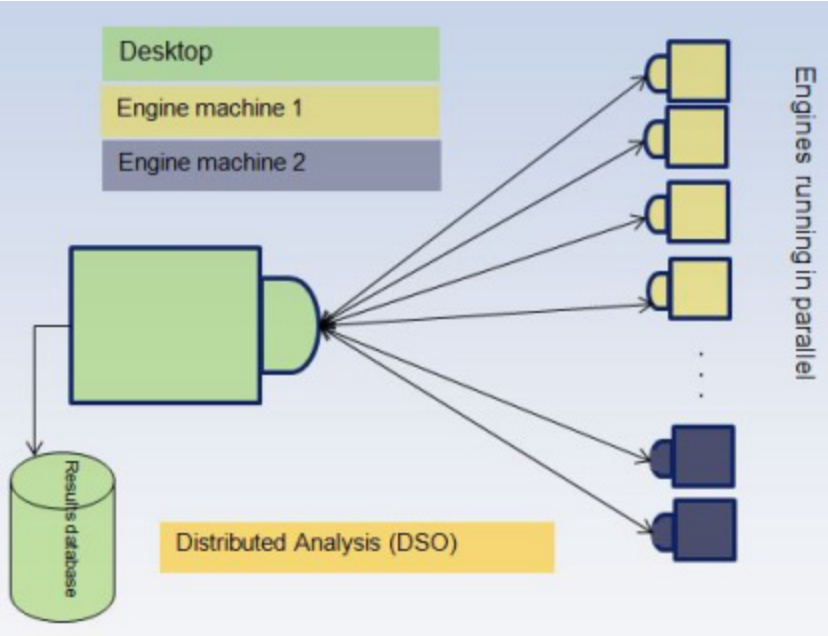
GM Specifics: the model used for 'Report-based extractor' jobs is NOT compatible with the 'ANSYS-extractor-for-GM' jobs. A valid model for ANSYS-extractor-for-GM cannot contain any of: reports, overlay plots, optimetrics calculations.

Related Topics

Large Scale DSO for Parametric Analysis

Large Scale DSO Theory

The parametric analysis command in Desktop computes simulation results as a function of model parameters, such as geometry dimensions, material properties and excitations. The parametric analysis is either performed on a local machine, where each variation is analyzed serially by a single engine, or distributed across machines through a DSO license. Desktop's DSO analysis runs multiple engines in parallel, thus generating results in a shorter time. In the Regular DSO algorithm, the parametric analysis job (Desktop) runs on master node, which in turn launches one or more distributed-parallel engines on each machine allocated to the job. Desktop distributes parametric variations among these engines running in parallel. As variations are solved, the progress/messages and variation results are sent back to Desktop, where they are persisted into the common results database on master node, as illustrated below:



A single engine can now span multiple machines with auto multi-level DSO.

Regular DSO Bottleneck

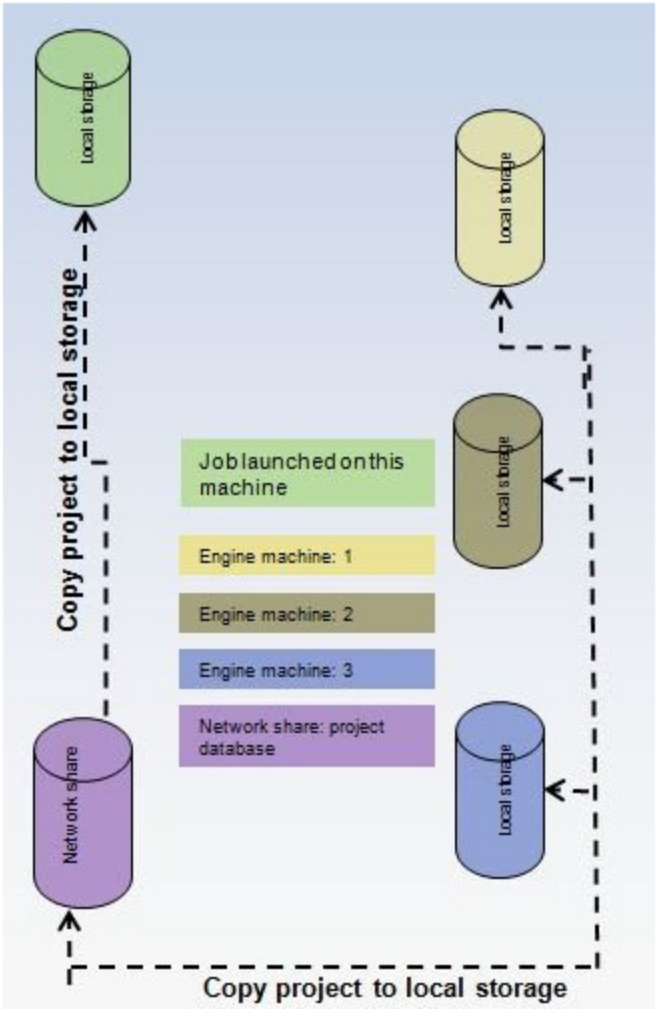
As per above illustration, Regular DSO's speedup is limited by the resources of the centralized 'Desktop' bottleneck. It's been observed that DSO becomes unreliable at a certain point, as the number of engines and number of variations is increased. The term ‘large scale parallel’ can be used to define this tipping point. For a given model, a ‘large-scale parallel’ job denotes scenarios, in terms of the number of distributed-parallel engines and the number of parametric variations, where the Regular DSO runs into centralized bottlenecks that result in one or both of: progressively smaller speedups, unreliability.

With the advent of economical availability and timely provisioning of compute resources, product designers have access to large compute clusters to run their simulations. And they are throwing larger and larger number of compute resources at simulation jobs, in order to obtain results faster. The parametric DSO needs to meet this challenge and target linear speedup for 'large-scale parallel' jobs. The Large Scale DSO feature is targeted toward 100% reliability and linear speedup of large-scale parallel DSO jobs.

Key Algorithms/Concepts for Large Scale DSO

- **Embarrassingly Parallel algorithm**: Large Scale DSO exploits the embarrassingly parallel nature of parametric runs. The solve of each variation is made fully independent of another variation, by replacing Regular DSO's centralized database with per-engine distributed databases
- **Distributed databases**: For each distributed-parallel engine, the database of the input model is cloned to local storage of engine's compute node (as illustrated in the following picture). Parallel analysis is performed on these cloned models and the analysis results also

go the corresponding location on local storage. As each engine has its own database, there is no ‘shared database’ contention between any two engines. There is negligible network traffic as analysis/computations are contained in a single machine and use just the ‘local resources’.



- **Hierarchical activation of engines**: In the Large Scale DSO algorithm, engine activation is done hierarchically, where the overhead of launching of engines is also shared across the nodes. Such hierarchical activation improves reliability of the activation phase of large-scale parallel jobs. A new 'desktopjob' program encapsulates hierarchical activation, as illustrated in the following pictures. In this approach, the 'root' desktopjob (Root DJ) activates a 'level-one' child desktopjob (DJ(L1)) for each unique node allocated to the DSO job. Each level-one desktopjob activates one or more 'leaf' desktopjobs (DJ(L2)) equal to the number of distributed engines per node. A leaf desktopjob in turn runs Desktop in batch mode, to perform local-machine parametric analysis to solve the variations assigned to this engine.
- **Decentralized Load balancing:** A parametric table is divided into regions of equal number of variations, with the number of regions equal to the number of engines. In the following illustration, the analysis of 30 parametric variations is distributed among 5 engines. Each

engine solves it's assigned region as a ‘local machine’ parametric analysis. Large Scale DSO job is considered done once all engines are done with their assigned variations.

- **Distributed results postprocessing**: As engine is done with the solve of a variation, it extracts results for the solved variation before progressing to the analysis of next variation. The extracted results are saved to the local storage. When the engine is done with analysis of all variations, the extracted results are transferred from local storage to the results folder of the input project.

Redistribution Feature

- Large Scale Distributed Solve Operation could submit a parametric setup to be solved in multiple machines, each machine may launch multiple EM-Desktop processes to solve the assigned variations(Design Points). Variations are distributed to each task(EM-Desktop process) equally, regardless of the machine hardware and each variation’s complexity. That may resultin some tasks finishing earlier than others, in some extreme case some tasks may hours behind fastest task.
- Now, redistribution occurs when a task finished its own assignment, it calls back to the L2 to ask for new assignment, L2 forward the request to L1, L1 forward it to L0, L0 may pick one of the slow task to remove some variations or return with no more assignment. When a task is picked to remove unsolved variation, L0 calls L1(may be different L1), L1 forward it to the selected L2, L2 makes a request to the EM-Desktop to remove some variations from its queue. EM-Desktop returns the removed variation indexes, or error code if it fails. In L0 if error is returned, it marks the selected task failed to respond the remove request, to void picking it again. L0 returns the result through L1-L2 back the EM-Desktop.
- The Ansys Electronics Desktop will enqueue the new assignment and then start solving those new variations. If the return data indicates error, it makes another call to L2 to get new assignment. If no more variations return, the Ansys Electronics Desktop finishs the simulation and exit.

Interactive Scheduler Jobs

This section includes information and guidelines for users running interactive scheduler jobs on Linux.

In most cases jobs run under a scheduler run as a batch job. These jobs may be submitted using the Ansys Electromagnetics Desktop job submission GUI, using cluster job submission commands on a command line or using a cluster GUI, if available.

An alternative method for submission of scheduler jobs can be used. For convenience, such jobs are referred to as “interactive scheduler jobs”. In this approach, the user submits an interactive job to the scheduler. From the interactive job prompt, the user launches an Ansys Electromagnetics Desktop product, which starts in interactive (GUI) mode, not batch mode. The user then selects a project and runs one or more analysis commands using the GUI. The intent is that these analysis commands should use all resources allocated to the job, whether on the same host as the GUI or on other hosts.

This approach is supported on Linux, where the user may set up an X Window System server for interacting with the Ansys Electromagnetics Desktop product GUI. The user needs to configure

the cluster environment and/or the interactive environment so that the user may view and interact with the product GUI. This approach is not supported on Microsoft Windows.

Related Topics

Specifying Options for Interactive Scheduler Jobs
DSO Configuration for Interactive Scheduler Jobs
Design Type Options for Interactive Scheduler Jobs

Specifying Options for Interactive Scheduler Jobs

Batch scheduler jobs use command line options for specifying options rather than GUI controls. For interactive scheduler jobs, options may be specified on the command line, using GUI controls, or obtained from the registry. The machines specified in the **Configurations** tab of the **HPC and Analysis Options** dialog will be ignored for interactive scheduler jobs. If the command line used to launch the product contains a list of specific machines (using option -machinelist list=...), then the job will use the specified machines and cores. If the command line used to launch the product does not contain a specific machine list, then the machines and cores allocated to the job by the scheduler will be used for the job. In general, the job distribution settings for interactive scheduler jobs should be specified on the command line, instead of using the **HPC and Analysis Options** dialog box.

- The UI shows the settings that will be used for analysis, even if they come from the command line, not the registry.
- If you make changes to the settings in the UI, and the changed settings will be used for the analysis, even if the changed settings conflict with the command line (including batchoptions).
- If you makes no changes to the settings in the UI, then the command line settings (including batchoptions) will be used for analysis.

Batchoptions settings which are automatically generated for batch jobs submitted using the Ansys Electromagnetics job submission GUI will need to be manually included in the product command line. Some use cases require that certain settings be made on the command line when the product is launched, rather than using the GUI.

Batchoptions for Interactive Scheduler Jobs

For interactive scheduler jobs, only a limited set of batchoptions are supported. These batchoptions include the DSO configuration options, the design-type-specific options, and the following additional design-type-specific options that are not currently listed in the user interface or command line help windows:

- **CreateStartingMesh:** Create the starting mesh only. Do not solve any other steps. Often used for multi-step jobs where the mesh creation is solved in a separate step using fewer resources than the rest of the analysis.
- **NumCoresPerDistributedTask:** Specifies the number of cores that are allocated to the job to use for each distributed task, used when running jobs under a scheduler. The scheduler

communicates the number of cores allocated to the job on each host. This setting is used to determine how to allocate the cores to each task.

- **RAMLimitPercent:** Specifies the percentage of system RAM that the analysis is expected to use on each host. Running out of memory can cause processes to be killed by the Operating System or other failures. The analysis will be gracefully terminated if it requires more than this limit. Typical limits are about 90%, to allow some RAM for the Operating System and other services. If there are multiple jobs running on a single host, then the RAM limit should be reduced accordingly. For example, if there are two jobs running on a host, then the limit could be 45%, which would allow 10% of the RAM for the Operating System and other services.
- **RAMLimitPerCoreInGB:** This is an alternative way to specify the RAM limit, instead of using the RAMLimitPercent. In this case, the RAM limit is the limit per core multiplied by the number of cores allocated to the job on each host. Typically used for jobs running under a scheduler to ensure that the scheduler memory limit and the ansysedt memory limit are consistent.
- **SolveAdaptiveOnly:** Solve adaptive passes only. Do not solve any other steps. Often used for multi-step jobs where the adaptive passes are solved in a separate step using fewer resources than the rest of the analysis.
- **TotalNumOfCores:** The total number of cores for the job. This option is only used for EKM (Engineering Knowledge Manager).
- **ValidateOnly:** Only validate the specified setup, design, or project. Do not analyze the specified setup, design or project.

Any other batchoptions will result in a warning message and will be ignored.

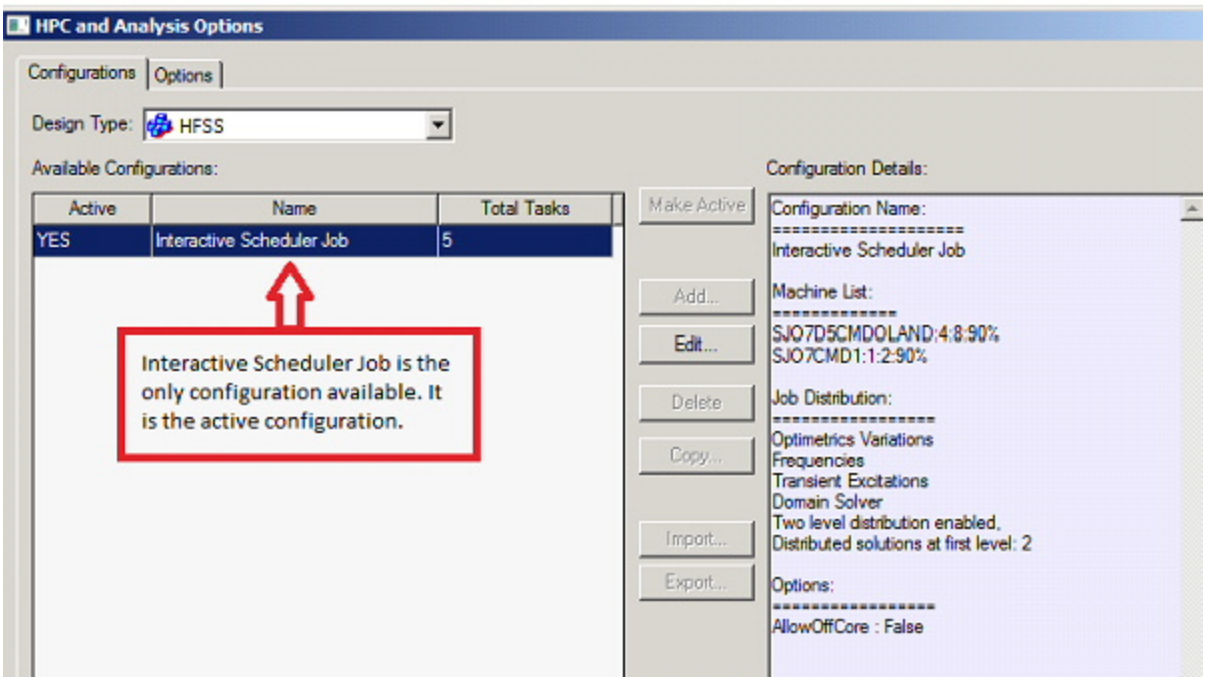
Command Line Example:

```
desktopjob.exe" -cmd dso -jobid RSM_27086 -machinelist
list=localHost:2:2:90%
-monitor -ng -batchoptions "
'LargeScaleDSO/MergeCsv'='acrossDPsAndTraces'"
-batchsolve TeeModel:Optimetrics:ParametricSetup2
E:\work\2018\LS_DSO\OptimTee.aedt
desktopjob.exe" -cmd dso -jobid RSM_25248 -machinelist
list=localHost:2:2:90%
-monitor -ng -batchoptions "
'LargeScaleDSO/MergeCsv'='acrossDPsByRow'"
-batchsolve TeeModel:Optimetrics:ParametricSetup2
E:\work\2018\LS_DSO\OptimTee.aedt
ansysedt -batchoptions " 'TempDirectory'='C:\\TEMP'
'HFSS/SelectedDSOConfiguration'='Local'
'Desktop/Settings/ProjectOptions/DoAutoSave'=1
'LargeScaleDSO/MaxFolderInMB'=100"
```

DSO Configuration for Interactive Scheduler Jobs

When running an interactive scheduler job, there is only one DSO configuration available for each design type. Each configuration is named “Interactive Scheduler Job”. This configuration is always the active configuration for an interactive scheduler job. and it is the only configuration displayed in the list of available configurations shown in the **HPC and Analysis Options** dialog box. No configurations can be added or removed, but the "Interactive Scheduler Job" configuration may be modified using the "Edit" button, which pops up the **Analysis Configuration** dialog.

Because the “Interactive Scheduler Job” configuration is the only configuration accessible for interactive scheduler jobs and it is not accessible in other modes, there is no sharing of the "Interactive Scheduler Job" configuration settings with other modes.



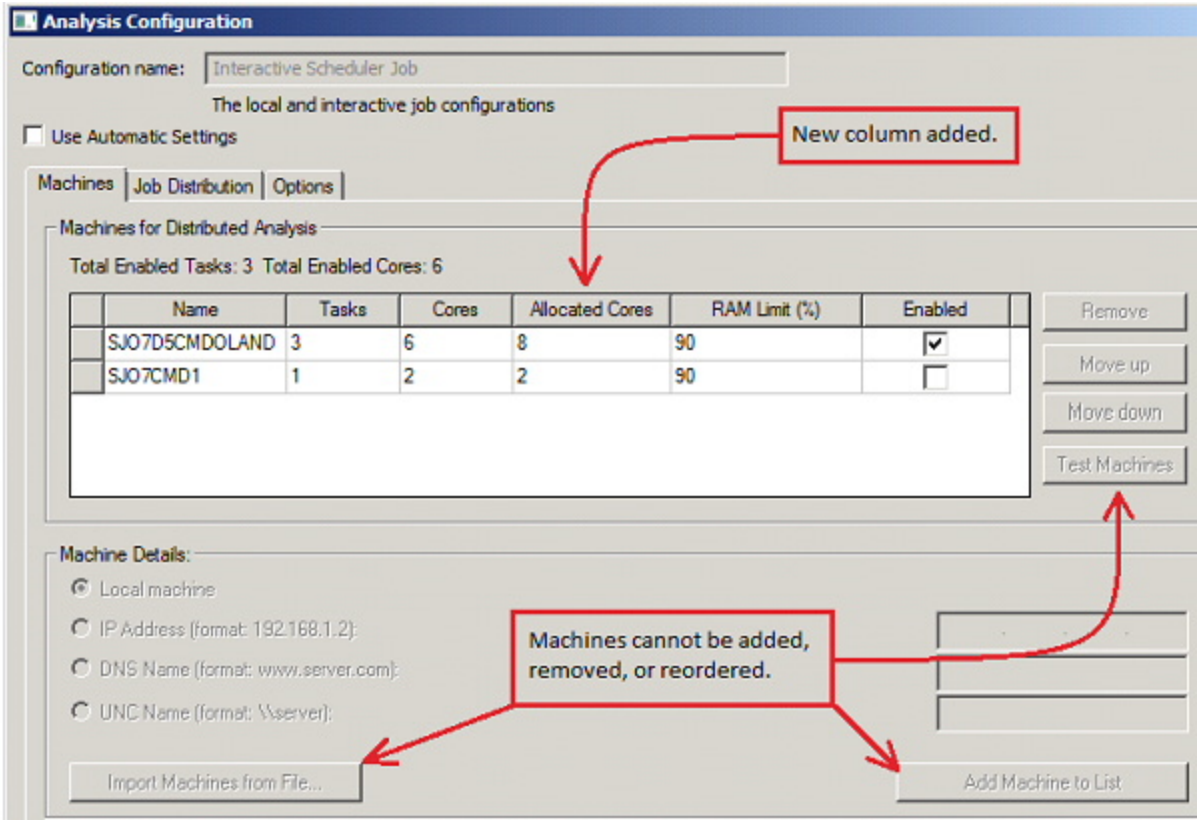
Major differences between interactive scheduler mode and batch mode or normal interactive mode are:

- In interactive scheduler mode, DSO configuration settings and design type options in the UI override command line options and batchoptions.
- When not in interactive scheduler mode, DSO configuration settings and design type options specified on the command line (including batchoptions) override UI settings.

Analysis Configuration - Machines Tab

The most obvious changes for interactive scheduler jobs are visible in the **Machines** tab of the **Analysis Configuration** dialog box. The grid of machine information is pre-populated with the list of machines allocated to the job. It also contains an additional column in interactive scheduler mode. This column, "Allocated Cores", indicates the number of cores allocated to the job by the scheduler on each host. You cannot add or remove machines from the list, or modify the allocated

cores for any machines. You can modify the tasks, cores or RAM limit for any machine, or specify that a machine is enabled or disabled.



Analysis Configuration - Job Distribution Tab

The **Job Distribution** tab of the **Analysis Configuration** dialog only appears if the "Use Automatic Settings" checkbox is not checked. The state of this checkbox, and the settings shown on this tab are initialized from the command line or from the registry. The command line options that affect the "Use Automatic Settings" checkbox or the job distribution settings are:

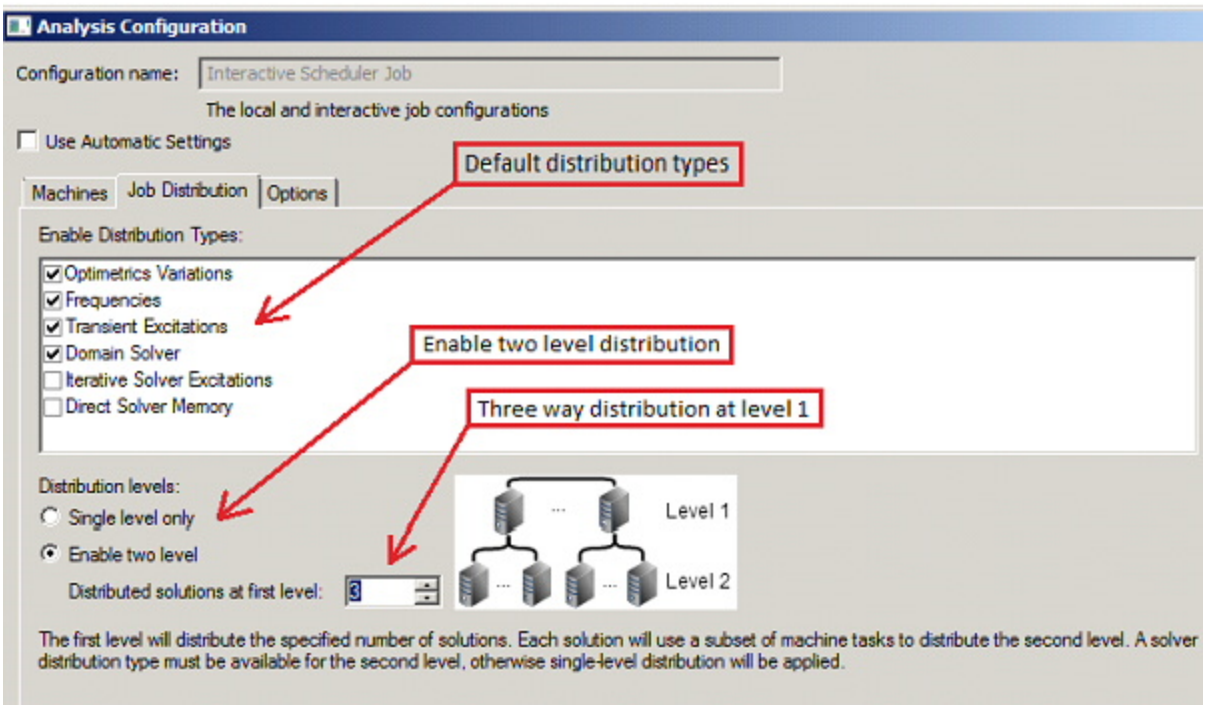
- -distributed
- -local
- -auto
- -machinelist

The registry contains the last value of these settings for an interactive scheduler job for the same user on the same host. Settings on the command line override settings from the registry. Any changes in the GUI will override the initial settings, even if the initial settings are from the command line. Any changes in the GUI also update the registry settings for the "Interactive Scheduler Job" configuration for the current design type.

Example

```
ansysedt -distributed includetypes=default maxlevels=2 numlevel1=3
```

Initial "Job Distribution" settings:



Analysis Configuration - Options Tab

You can use the **Options** tab of the **Analysis Configuration** dialog to examine or modify the DSO configuration options. The DSO configuration options are handled like the design type options, except that the **Interactive Scheduler Job** configuration settings are not shared with other modes.

The settings on this tab are initialized from the command line (using the `-batchoptions` command line option) or from the registry. The registry contains the last value of these settings for an interactive scheduler job for the same user on the same host. Settings on the command line override settings from the registry. Any changes in the GUI will override the initial settings, even if the initial settings are from the command line. Any changes in the GUI also update the registry settings for the "Interactive Scheduler Job" configuration for the current design type.

Example

```
ansysedt -batchoptions 'HFSS/AllowOffCore'=0
```

Design Type Options for Interactive Scheduler Jobs

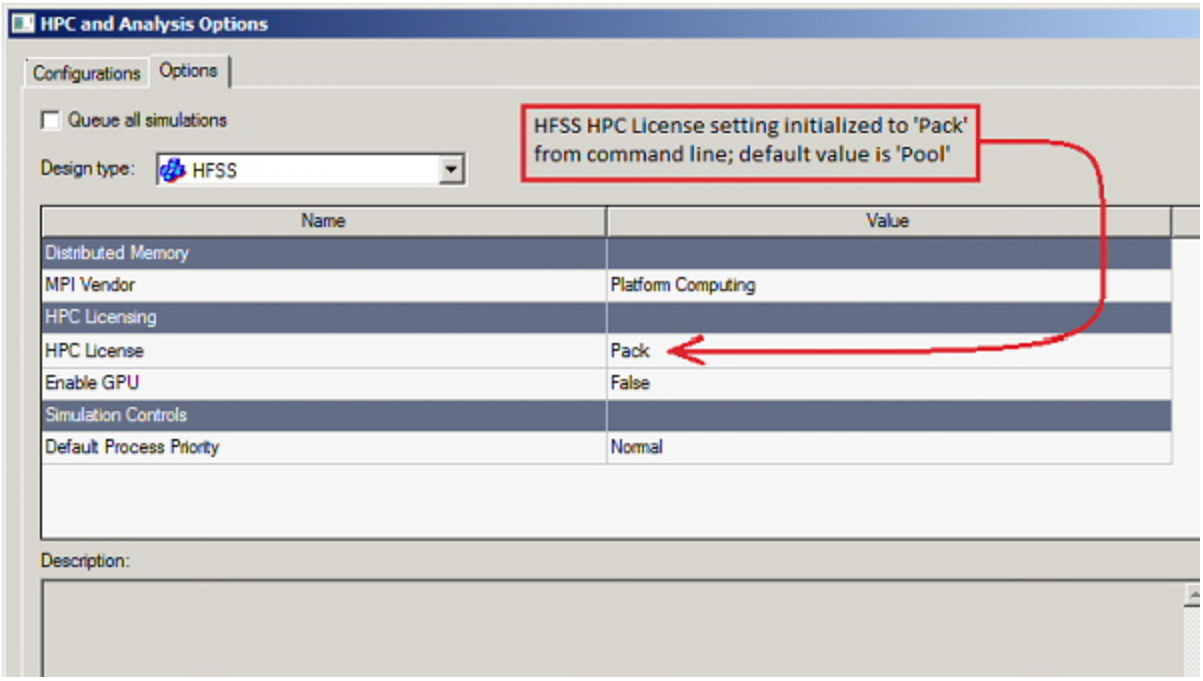
You can use the Options tab of the HPC and Analysis Options dialog to examine or modify the design type options. The settings on this tab are initialized from the command line (using the -batchoptions command line option) or from the registry. The registry contains the last value of these settings for the same user on the same host. Settings on the command line override settings from the registry. Any changes in the GUI will override the initial settings, even if the initial settings

are from the command line. Any changes in the GUI also update the registry settings for the current design type.

These settings are shared between interactive scheduler mode and other modes.

Example:

```
-batchoptions 'HFSS/HPCLicenseType'='Pack'
```



Distribution Command Line Options

The user should include options specifying how the job should be distributed in the command line. These command line options will override options specified using the **HPC and Analysis Options** dialog box or obtained from the registry. Refer to Running Maxwell From a Command Line for more information on command line options.

For both scheduler batch jobs and interactive scheduler jobs, the -MachineList num=<num distributed tasks> format is the most common way to specify the number of tasks for the job. The other formats (-MachineList list=... or -MachineList file=...) allow the user to specify the number of tasks and cores to use on each host. These formats may be useful with clusters of heterogeneous machines, by allowing the user to specify different numbers of tasks or cores for different hosts. If either of the latter two formats is used, the user must ensure that the hosts and cores specified on the product command line are compatible with the hosts and cores allocated to the job.

Batchoptions for Interactive Scheduler Jobs

For interactive scheduler jobs, only a limited set of batchoptions are supported. These batchoptions include the DSO configuration options, the design-type-specific options, and the

following additional design-type-specific options that are not currently listed in th user interface or command line help windows:

- **CreateStartingMesh:** Create the starting mesh only. Do not solve any other steps. Often used for multi-step jobs where the mesh creation is solved in a separate step using fewer resources than the rest of the analysis.
- **NumCoresPerDistributedTask:** Specifies the number of cores that are allocated to the job to use for each distributed task, used when running jobs under a scheduler. The scheduler communicates the number of cores allocated to the job on each host. This setting is used to determine how to allocate the cores to each task.
- **RAMLimitPercent:** Specifies the percentage of system RAM that the analysis is expected to use on each host. Running out of memory can cause processes to be killed by the Operating System or other failures. The analysis will be gracefully terminated if it requires more than this limit. Typical limits are about 90%, to allow some RAM for the Operating System and other services. If there are multiple jobs running on a single host, then the RAM limit should be reduced accordingly. For example, if there are two jobs running on a host, then the limit could be 45%, which would allow 10% of the RAM for the Operating System and other services.
- **RAMLimitPerCoreInGB:** This is an alternative way to specify the RAM limit, instead of using the RAMLimitPercent. In this case, the RAM limit is the limit per core multiplied by the number of cores allocated to the job on each host. Typically used for jobs running under a scheduler to ensure that the scheduler memory limit and the ansysedt memory limit are consistent.
- **SolveAdaptiveOnly:** Solve adaptive passes only. Do not solve any other steps. Often used for multi-step jobs where the adaptive passes are solved in a separate step using fewer resources than the rest of the analysis.
- **TotalNumOfCores:** The total number of cores for the job. This option is only used for EKM (Engineering Knowledge Manager).
- **ValidateOnly:** Only validate the specified setup, design, or project. Do not analyze the specified setup, design or project.

Any other batchoptions will result in a warning message and will be ignored.

Example:

```
ansysedt -batchoptions " 'TempDirectory'='C:\\TEMP'
'HFSS/SelectedDSOConfiguration'='Local'
'Desktop/Settings/ProjectOptions/DoAutoSave'=1
'LargeScaleDSO/MaxFolderInMB'=100 "
```

Batchoptions

Batchoptions may be specified in the command line used to launch the product. Any batchoptions specified in the command line will override the associated registry settings. Batchoptions will also override options specified using **HPC and Analysis Options** window or other dialog boxes used to specify options.

Setting the Number of Cores per Distributed Task

When submitting a job using the Ansys Electromagnetics job submission GUI, the number of cores per distributed task for a job is specified using the batchoption with pathname '<*DesignType*>/NumCoresPerDistributedTask', where <*DesignType* is the design type to analyze. The batchoption setting is automatically included in the product command line when the job is submitted to the scheduler.

For interactive scheduler jobs, the user must include the associated batchoption setting or settings in the product command line when the product is launched. Multiple batchoption settings are required if the user analyzes multiple design types using the same product process. Batchoptions are the only way to specify this setting for batch jobs. There is an alternative to using the NumCoresPerDistributedTask batchoption for Interactive Scheduler Jobs. The user may specify the total number of tasks and the total number of cores for each machine using the **Machines** tab of the **Analysis Configuration** window. You may use **Edit** in the **HPC and Analysis Options** to open the **Analysis Configuration** window for the "Interactive Scheduler Job" configuration.

Setting the Ram Limit Per Core in GB

When cores and RAM per core are requested, the cores could be allocated in an arbitrary/non-uniform way across nodes that themselves could be non-uniform/heterogeneous. For example, nodes could range from 4 cores to 20 cores and from 64 GB to 384 GB. In such environments, the RAM percentage set in the HPC configuration, Job Submission Compute Resources, or batch option may not be appropriate. In such cases, you can use the batchoption for RAMLimitPerCoreInGB. This setting specifies the Maximum amount of RAM used for each core allocated by the scheduler in GB. This batchoption cannot b e combined with RAM limit percent and is only valid when solving in a Linux scheduler environment.

The scheduler GUI automatically passes this new batch option instead of percent limit. From a scheduler GUI, such a request is available only for auto.

You can also use this new batch option for command line submission. The desktop does the computations and passes the percent limit to product/solver.

Setting the Remote Spawn Command Option to Scheduler

The Remote Spawn Command setting is only meaningful when running on the Linux Operating System. The value 'Scheduler' is valid if the job is a scheduler job running under an LSF or SGE scheduler, and only if the MPI Vendor is 'Intel'.

When submitting a job using the Ansys Electromagnetics job submission GUI, the Remote Spawn Command for an analysis may be specified using the batchoption with pathname *DesignType/RemoteSpawnCommand*, where *DesignType* is the Design Type to analyze. The Remote Spawn Command setting is only meaningful when running on the Linux Operating System. The value “Scheduler” is valid if the job is a scheduler job running under an LSF or SGE scheduler, and only if the MPI Vendor is “Intel". To specify the value “Scheduler” for this option for an interactive scheduler job, the Remote Spawn Command must be specified using the

DesignType/RemoteSpawnCommand batchoption in the product command line when the product is launched. In addition, the *DesignType/MPIVendor* batchoption must be specified with value “Intel” in the product command line when the product is launched. For interactive scheduler jobs, the Remote Spawn Command and the MPI Vendor may be specified with batchoptions or as design type options in the **HPC and Analysis Options** dialog box.

Setting MPI Version

When submitting a job using the Ansys Electromagnetics job submission GUI, the MPI Version may be specified using the batchoption with pathname DesignType/MPIVersion, where DesignType is the type of design to analyze (e.g. HFSS). It allows selection of which Intel MPI version to use, for both Windows and Linux. Valid values are "Default", "2018", and "2021".

For interactive solves, the MPI version may be specified with the batchoption in the command line used to launch the product, or as a design type option in the HPC and Analysis Options dialog box.

High Performance Computing (HPC) Integration

Ansys Electromagnetics products offer a direct integration with a number of High Performance Computing (HPC) software programs. This direct integration does not require Ansys Electromagnetics RSM Service. The list of currently-supported HPC software includes:

- Ansys Cloud
- Platform’s Load Sharing Facility (LSF)
- Microsoft Windows® HPC Server
- PBS Pro or PBS Torque
- Grid Engine (GE)

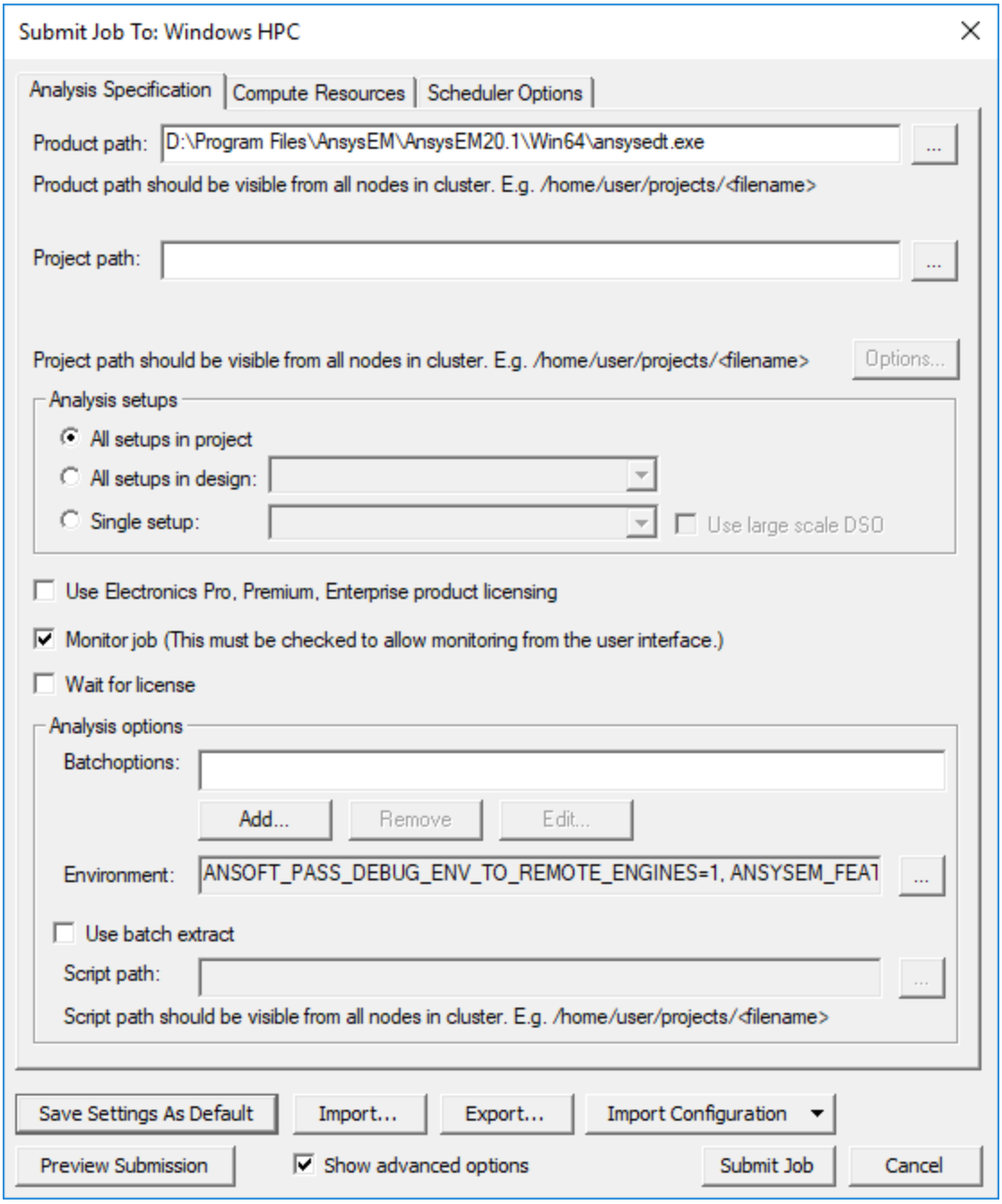
You can also do custom integration.

Note	For additional information about high performance computing not in this guide, see the *Ansys Electromagnetics HPC Administrator's Guide* in the help.

A job scheduler may also be described as a batch system, a Distributed Resource Management System (DRMS) or Distributed Resource Manager (DRM). The features supported on each scheduler are included in the documents for each. For each job scheduler, the versions or revisions that have been tested are included.

HPC Computing integration for Ansys Electronics Desktop supports Lustre FS. However, Ansys doesn't support having Linux home directories stored on Lustre FS. (This is a limitation due to the way MainWin writes temporary/config files into the home directory.)

A user may submit jobs using the command line tools or other tools provided by the scheduler. The Desktop includes a GUI to help the user submit jobs to a job scheduler. This generic Job Submission GUI is shared across the Ansys Electromagnetics products.



The general procedure is to specify the scheduler and head node, describe and submit the job, and monitor the results.

The **Submit Job To** dialog contains three tabs:

- **Analysis Specification**--specify the Product path, Project name, the setups, and analysis options such as batchoptions, or, for advanced users, Environment variables. The Project can be an archive. The project file pathname must be a UNC path that is accessible from

each compute host used for Ansys Electromagnetics jobs.

The Product path and Project fields support mapped drives. This is done through a checkbox in the Specify Product Path dialog displayed when you click the ellipsis [...] button. A checkbox lets you "Use converted UNC path if mapped drive specified." If you select a project or product on a mapped drive, and check the option, the converted UNC path equivalent to the mapped drive pathname is used.

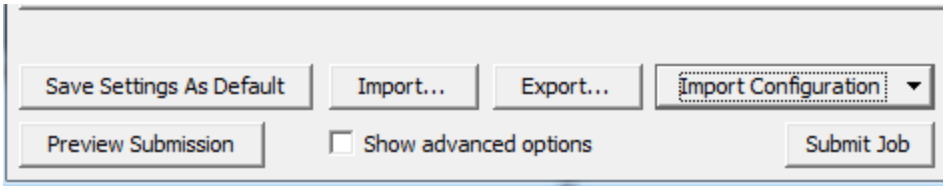
- **Compute Resources**--specify whether to use automatic settings, resource selection parameters, and if, you do not use automatic settings, job distribution parameters.
- **Scheduler Options**--specify for Job name and priority.

There are two ways that the GUI may be used to submit jobs.

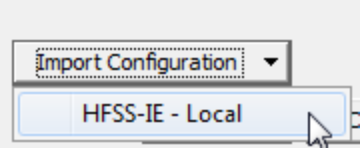
- The first mode requires that the Desktop (UI) process run on a host which is also a submission host for the job scheduler. This mode is called local mode or working mode.
- The second mode is useful for cases in which the submission hosts are not able to run graphical processes, such as the Desktop. The second mode is only supported on Linux in the Ansys Electromagnetics Suite 2023 R1 release. In the second mode, an administrator configures the RSM Service to act as an interface to the job scheduler, and starts the RSM Service on a submission host for the cluster. The user runs the Desktop (UI) process on another host (which may be called the postprocessing host). To submit a job, the user specifies the host where the RSM Service is running, and the Desktop process connects to the RSM Service over the network to submit the job. In this mode, some configuration is required, and the RSM Service typically must run as a privileged user (e.g., root), so that it can launch processes as any user.

Import and Export Configurations

The bottom of the Job Submission GUI has buttons for Import..., Export..., and Import Configuration let you save a configuration for each solver type.



- The **Export** button exports most of the settings of this dialog (all tabs) to a file.
- The **Import** button updates most of the settings in this dialog (all tabs) from a file.
- The **Import Configuration** button updates the DSO settings in this dialog from any DSO configuration as shown in the **Configurations** tab of the **HPC and Analysis Options** dialog box. The **Design Type** of the DSO configuration must match the design type of one of the designs in the project, so the "Project" must be specified before using the **Import Configuration** button. The batchoptions are also set from the specified DSO configuration or from the Design Type options settings, which are shown in the **Options** tab of the **HPC and Analysis Options** dialog.



The **Export** and **Import** buttons may be used to save and then restore a frequently used collection of job submission settings. The **Save Settings as Default** button may also be used to save the current settings, but it always overwrites any previously saved settings. Using the **Export** button, the user may save multiple sets of settings, or may transfer the settings to another machine.

The **Select Scheduler** dialog also has **Export** and **Import** buttons. These buttons may be used to save the settings in this dialog to a file or restore them from a file.

An Exported configuration is named **Submit_Job_Settings** by default and has an **.areg** suffix. A file browser window opens in the project folder and lets you name an exported file and location, and select an **.areg** file to import.

Related Topics

Scheduler Terminology

What a Scheduler Does

Installation of Ansys Electromagnetics Tools

Ansys Electromagnetics Jobs

Submitting and Monitoring Ansys Electromagnetics HPC Jobs

Job Submission Scripting

Scheduler Terminology

- Core: unit of processing
- Processor: consists of one or more cores
- Machine/Host/Node: consists of one or more processors, memory, disk, etc.
- Resource: Machines, licenses, etc. that are used by a Job
- Job: Application (also called: program, executable), with command line options, that uses resources to produce useful results. For example, maxwell.exe –ng –BatchSolve …
- Serial Job: job that runs on a single core
- Parallel Job: job that runs on multiple cores (belonging to same or different machines)
- Compute Cluster: network of machines on which Jobs run. Typically, consists of head node (s) and many compute nodes
- Service: Program that runs in the background (e.g., RSM Service). ‘Listens’ on a ‘port’. OS provides programming interface by which Applications communicate with service, once machine and port number are known. Launching an executable on remote machine, requires a service to run on remote machine.

Ansys Electromagnetics Suite Terminology

- Desktop: The main application used to accomplish a task, such as Maxwell. The desktop may run as a GUI or it may run as a batch command.
- Engine: Application (the executable) that is launched by analysis commands, to generate analysis results.
- Multi-processing: A single engine uses multiple cores on the **same** machine.
- Distributed-processing: Multiple engines are launched simultaneously (on same machine **or** different machines). Uses ansoft_distrib (and related) license.

Related Topic

What a Scheduler Does

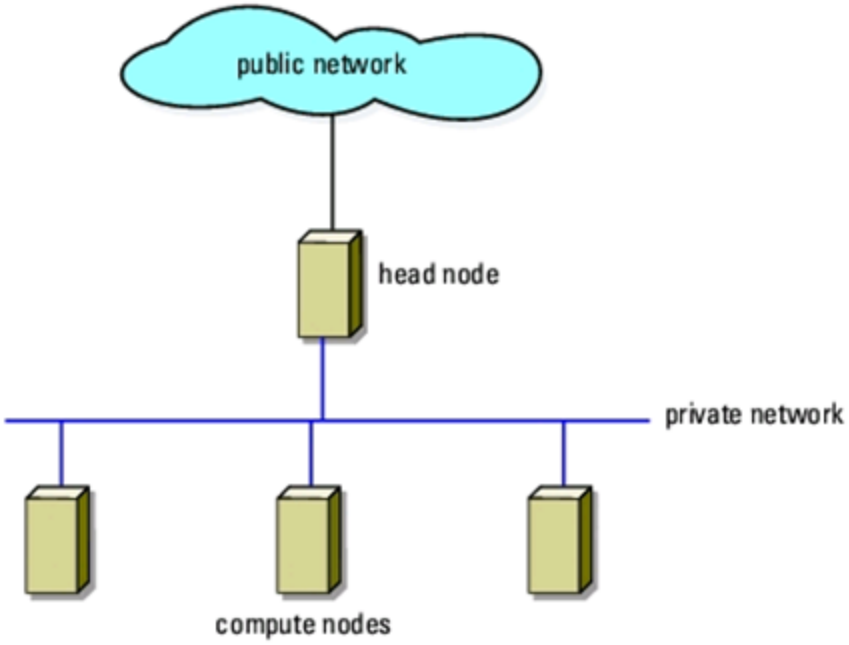
Command Line Enhancements for Ansys Electromagnetics Suite Desktop Products

High Performance Computing (HPC) Integration

What a Scheduler Does

- Enables effective/efficient utilization of cluster's resources consistent with organization's goals
- Maintains queue(s) of jobs
- Maximizes throughput of the jobs by processing all jobs as fast as possible
- Typically, one job per cpu policy
- Allows choice of various scheduling policies (e.g. First Come First Serve, Priority Based, Preemption)
- Provides a suite of tools or utilities (graphical or command line) for end user to submit jobs, monitor jobs, abort jobs, suspend jobs, …
- Manages a compute cluster by running various interacting 'services' on head nodes and compute nodes
- Provides a programming interface to access 'services'

Scheduler Managed Compute Cluster



Head node(s) typically maintains queues. Compute nodes are typically on a high speed network, to improve scalability of parallel jobs. Services running on nodes interact with each other to manage resources. End user tools communicate with services to submit/abort/suspend/etc. jobs.

Related Topic

High Performance Computing (HPC) Integration

Command Line Enhancements for Ansys Electromagnetics Desktop Products

Installation of Ansys Electromagnetics Suite

Ansys Electromagnetics Suite must be available on each cluster host where jobs may be run.

- On a LINUX platform, Ansys Electromagnetics Suite may be installed on a shared drive, that is accessible to all machines in the cluster.
- On a Windows platform, Ansys Electromagnetics Suite must be installed separately on each host of the cluster.

The Ansys Electromagnetics Suite must be accessible using the same path on each host. All cluster users running Ansys Electromagnetics jobs must have permission to read and execute the files in the installation directory and its subdirectories.

The Temp directory selected during installation must be readable and writable by all user accounts used to run the Ansys Elctromagnetics Suite. This temp directory path should be the same on all machines of the cluster and should be local to every machine. For example, c:\temp on Windows, /tmp on LINUX

Because HPC is offered as a direct integration, you need only install the Ansys Electromagnetic Suite software. No additional configuration is required.

Example

Install the Ansys Electromagnetics Suite in directory C:\Program Files\AnsysEM\ on each node of the cluster. The same directory pathname must be used on all hosts.

Related Topics

High Performance Computing (HPC) Integration

Firewall Configuration

Installation Directory Examples

Firewall Configuration

If firewall is turned OFF between the machines of the cluster, there is no need for any configuration. If firewall is turned ON, you, or a system administrator, should perform the steps below.

- Windows cluster: Configure firewall by adding exceptions that allow Ansys Electromagnetics Suite programs and services to communicate with each other. If you are using standard Windows Firewall, this is automatically done for you, by the Ansys Electromagnetics installation program. On the other hand, if you are using a 3rd-party firewall software, it needs to be configured in a similar manner.
- Linux cluster: Open up the firewall for range of ports denoting ephemeral (or dynamic) ports. Check with your system administrator on how this can be done on each machine of cluster.

Related Topics

High Performance Computing (HPC) Integration

Installation Directory Examples

Using Ansys EM HPC Diagnostics tool

Installation Directory Examples

Microsoft Windows Example

Install the Ansys Electromagnetics Suite in directory C:\Program Files\AnsysEM\v231\win64 on each node of the cluster. The same directory pathname must be used on all hosts.

Linux Example

Install the Ansys Electromagnetics Suite in a common directory that is accessible using the path: /opt/AnsysEM/v231/ on each execution node of the cluster.

Related Topics

High Performance Computing (HPC) Integration

Firewall Configuration

Ansys Electromagnetics Jobs

Using Ansys EM HPC Diagnostics tool

Integration with Microsoft Windows® HPC Scheduler

The Windows HPC scheduler is only supported on Windows. Jobs may be submitted in any of the following ways:

- Using the Windows HPC GUIs from Microsoft: Job Manager or Cluster Manager
- Using the Windows HPC command line tools (job, etc.)
- Using the Desktop UI commands for Scheduler selection, Job submission and Job monitoring/control. You specify the Windows HPC Scheduler User Interface for Submit Job by clicking **Tools>Job Management>Select Scheduler...**

You can also select the **Simulation** tab of the ribbon, and click the **Scheduler** icon.

Select the scheduler from the drop-down menu on the **Select Scheduler** window. Once you select a scheduler, you can access the interface for job submission, monitoring and control.

See the Ansys Electromagnetics Suite Windows Installation Guide for additional information on supported schedulers.

General Guidelines for Submitting Ansys Electromagnetics Jobs

A Job submitted to Windows HPC Cluster is defined by Job properties, Task List and Task properties. Priority, resource requirements, node preferences, etc. come from Job properties. In the case of Ansys Electromagnetics jobs, Task List consists of a single task. Properties of this task specify command line that runs Ansys Electromagnetics Desktop in non-graphical mode to perform analysis of an Ansys Electromagnetics project.

Specifying the Number of Compute Resource Units for HPC Jobs

In the old setup you had to list machines multiple times, and figure out how many cores per task on each machine in order to set Number of Processors Distributed to the smallest of these. In the new setup, you just enter the number of tasks and total cores per machine.

Ansys Electromagnetics Project File and Project Directory for use with Windows HPC Scheduler

Ansys Electromagnetics Suite tools write their results to a subdirectory of the directory containing the Ansys Electromagnetics project file. The Project Directory (the directory containing the project file) must be accessible to all of the cluster hosts that may run Ansys Electromagnetics jobs. The user account for the job must have permission to read the project directory, and to create and modify files and subdirectories of this directory. The pathname of the project file must be

accessible to all cluster hosts using the same path name, which is generally expressed as a UNC pathname.

Example:

The project file is on the user's workstation (with hostname user1_PC) in directory C:\user1\projects\new\project1.aedt, and the directory C:\user1\projects is shared with sharename projects.

Correct

When submitting the job, you should use the following pathname to specify the project file:

```
\\user1_PC\projects\new\project1.aedt
```

Incorrect

If a local pathname is used, the cluster hosts will not be able to find the user's project on the workstation

```
user1_PC: ' C:\user1\projects\new\project1.aedt'
```

Related Topics

Windows® HPC Job Templates

Selecting Computation Resource Units (Job Unit Type)

Windows® HPC Job Credentials

Submitting and Monitoring Ansys Electromagnetics HPC Jobs

Jobs may be submitted to the Windows HPC Scheduler using any of the following methods:

- Using the **Submit HPC Job** dialog
- Using the Windows HPC Job Manager GUI
- Using the Windows HPC Command Line Tools
- Using the Windows PowerShell

Client Utilities from the Microsoft HPC Pack, must be installed on the submit host to use any of these methods to submit a job to a cluster. The **Submit HPC Job** dialog will be unable to contact the cluster head node if the client utilities are not installed.

This document covers the first method. See the Microsoft documentation for information on the other three methods.

- Submitting and Monitoring Jobs for Windows HPC
- Specifying the Number of Compute Resource Units for HPC Jobs

Jobs may be submitted from any Microsoft Windows host meeting the following requirements:

- For submitting jobs to the Windows HPC scheduler, the Desktop process must run on a node that is configured for submission of jobs to the Windows HPC cluster. That is, the Windows HPC Client Utilities must be installed on the node, and network communication from the Desktop node to the head node of the cluster must be allowed. For Ansys Electromagnetics Suite 2023 R1, Windows HPC Server 2008 R2 (or later) client utilities are required. Using a computer on the network is not supported for submission of jobs to the Windows HPC cluster.

- When submitting jobs to a Windows HPC cluster, the user must also specify the head node of the cluster to which the jobs will be submitted. When the user selects the "Windows HPC" scheduler in the "Choose scheduler" list, the Head Node edit control is enabled. The user may enter the Windows HPC cluster head node name into the edit box. Alternatively, the head node may be selected using a "Browse for Computer" browser by pressing the "..." button.
- The Windows HPC Pack client utilities are installed on the submission host.
- Network communication between the submission host and the Windows HPC Cluster head node is permitted; there is a network connection between these hosts that is not blocked by any firewall or the like.
- The submission user is permitted to submit jobs to the Windows HPC Cluster.

Job Monitoring

Windows HPC Jobs may be monitored using the **Monitor Job** dialog which is brought up by the **Tools > Job Management > Monitor Jobs...** command. This dialog may also be brought up by checking the **Begin monitoring this job now** checkbox when a job is successfully submitting using the job submission dialog box. In addition to the above requirements to allow job monitoring the following is also necessary:

- Network communication between the submission host and all Windows HPC Cluster nodes where the job may run is permitted; there is a network connection between these hosts that is not blocked by any firewall or the like

Cluster Configuration

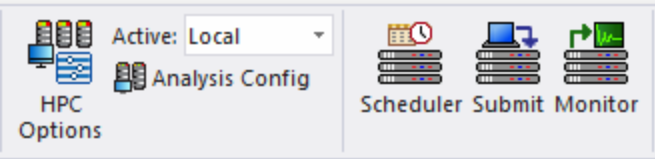
Any job running on a Windows HPC Cluster that is distributed over multiple compute hosts requires network communication between processes running on these hosts. The cluster must be configured to allow this communication. Any firewall or other security software must be disabled or configured to allow communication between any of the compute hosts were a job could run.

Job Submission User Profile on Cluster Compute Nodes

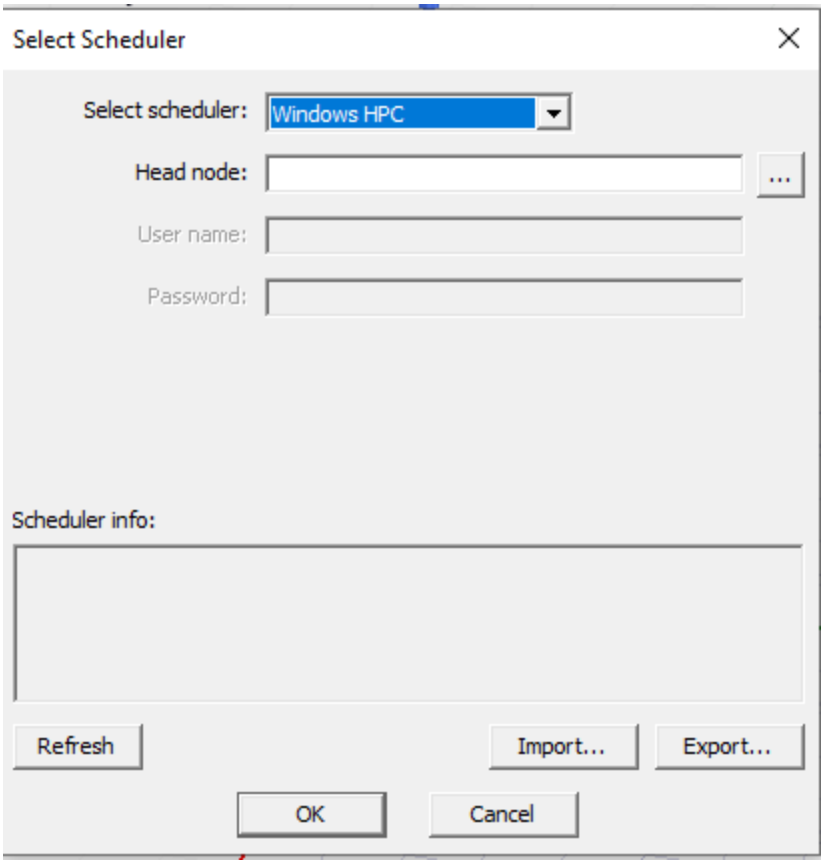
In order for a job to run correctly, the submission user's profile must be accessible and properly initialized on the cluster compute nodes where the job runs. If the Ansoft/temp subdirectory of the user's "My Documents" directory does not exist or is not accessible on the compute cluster nodes where a job runs, the batchoptions for the job will not be processed correctly, resulting in job failure. One way to ensure that this directory is created on each compute host is for the submission user to login to each compute host and run the product GUI one time.

Submitting and Monitoring Jobs for Windows HPC

In order to submit jobs using **Windows HPC**, you must click **Tools>Job Management> Select Scheduler** or you select the **Simulation** tab of the ribbon, and click the **Scheduler** icon.



This opens the **Select Scheduler** dialog box. Specify Windows HPC as the scheduler.



For Windows HPC, specify the head node of the cluster.

After specifying the cluster head node, you can click **Refresh**. This verifies that the head node may be contacted, and displays the scheduler name, a brief description (including the head node name), and the version of the Windows HPC head node.

Pressing **Cancel** discards changes made in this dialog box. Pressing **OK** verifies that the head node can be contacted before accepting the changes. If no problem occurs, the dialog will be closed. If there is a problem contacting the head node, the dialog will not be closed and the changes are not accepted.

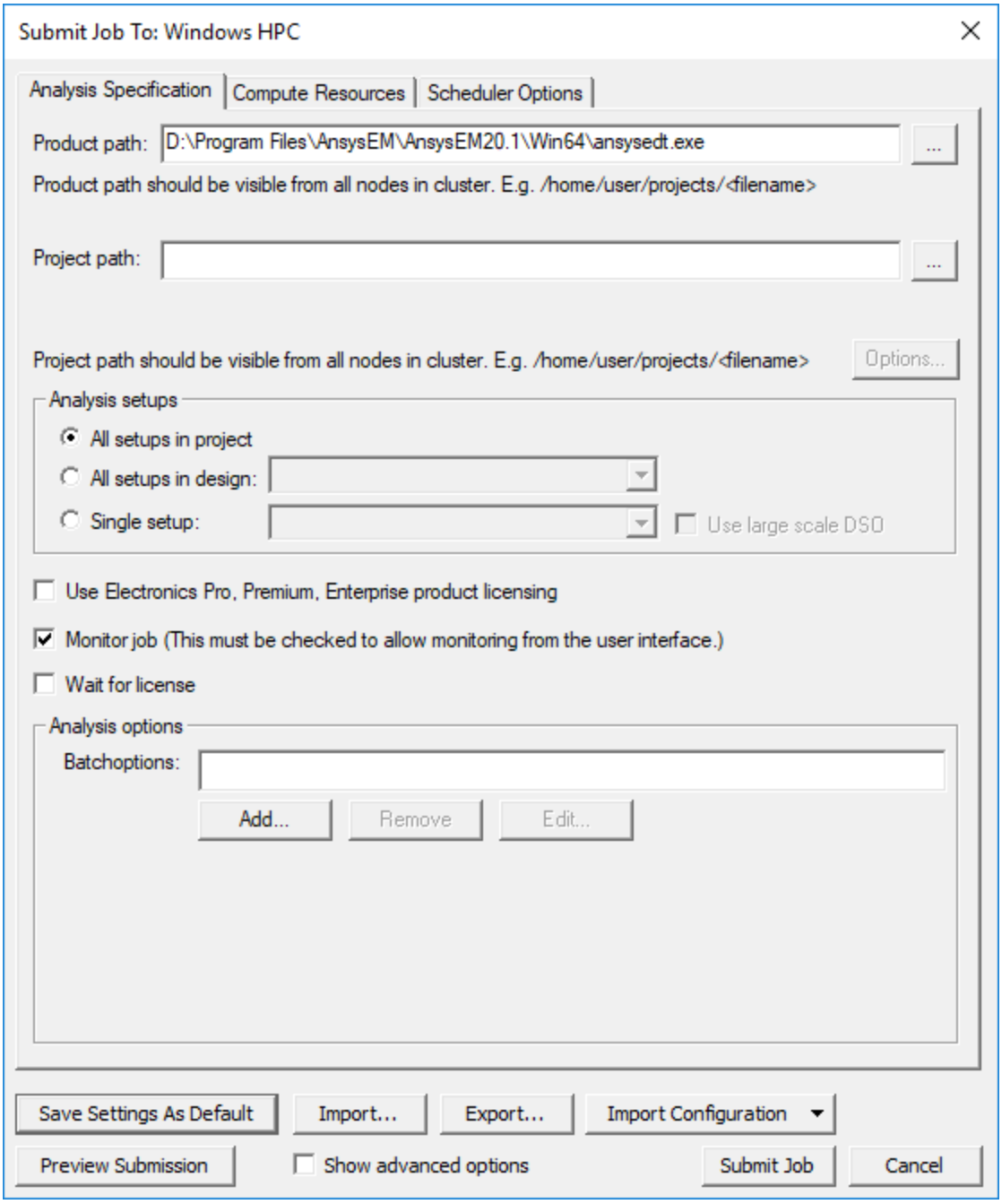
After setting the job submission node, select **Tools> Job Management> Submit Job...** or **Project> Submit Job...** or [ProductName]**> Submit Job...** to open the *Submit Job To:* dialog box. You can also select the **Simulation** tab of the ribbon and click the **Submit** icon. You can also

access **Submit Job** from the shortcut menus for the Project Name, Design name, Analysis Setup, or Optimetrics Setup.

The Submit Job To: dialog: contains three tabs:

- **Analysis Specification**--specify the Product path, Project name, the setups, and analysis options such as batchoptions, or, for advanced users, Environment variables. If you select the Analysis or Optimetrics setup, the Analysis Specification is pre-populated.
- **Compute Resources**--this tab can be populated either by automatic settings, by predefined Analysis Configuration, or specifying parameters in the fields for resource selection, for job parallelization and enabled forms of parallelization.
- **Scheduler Options**--contains fields for Job name and priority. The customization options shown by checking advanced are not used for Windows HPC.

In the **Analysis Specification** tab, enter the pathnames of the product path and of the project file in the “Project” edit box. These must be UNC paths that are accessible from each compute host used for Ansys Electromagnetics jobs. The Project can be an archive. The submission user must have permission to write to the directory containing the project file.



You can select which setups are analyzed in the Analyze Setups section of this dialog box. There are radio buttons to select:

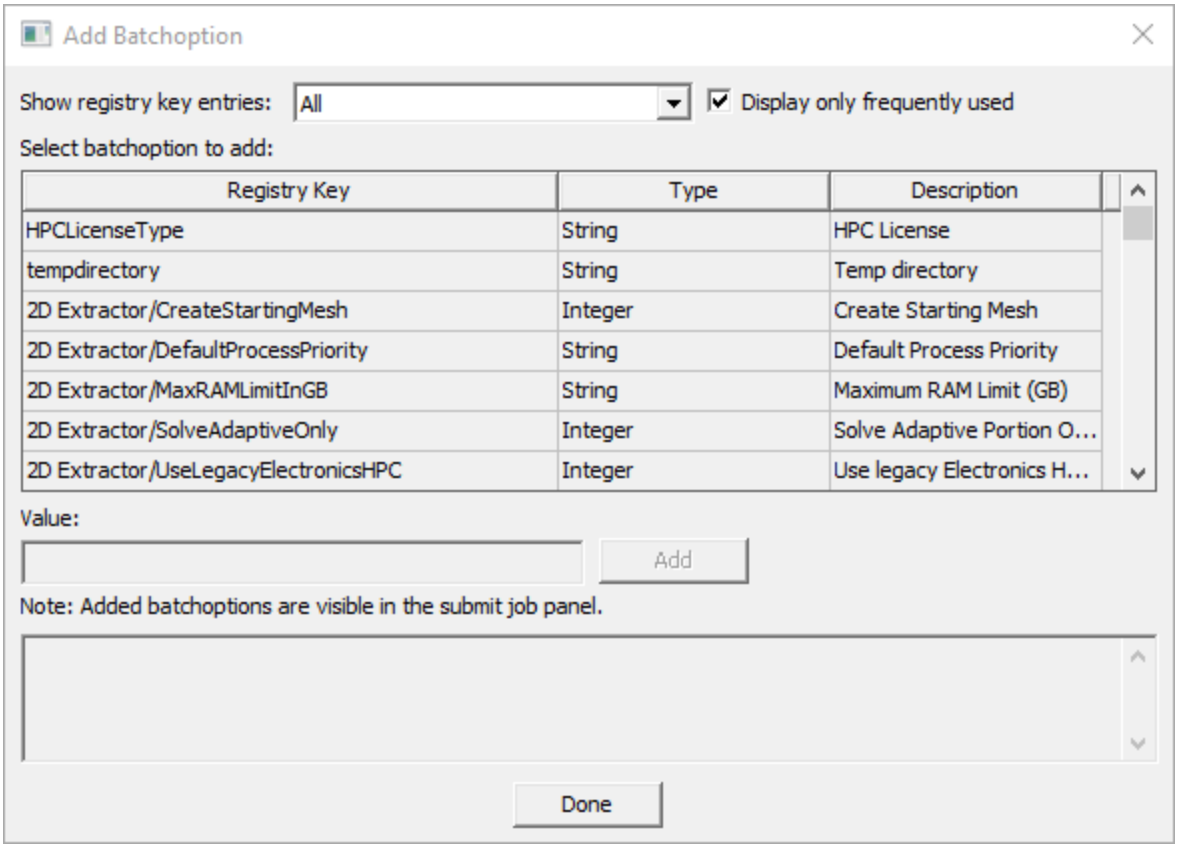
- All setups in the project
- All setups in a specified design: you select the design from the drop-down menu
- Single setup. If you select the Submit Job command from the shortcut menu, the setup name populates the field.

If you specify multiple setups, they will be processed sequentially in the order displayed in the edit box.

The Analysis options include:

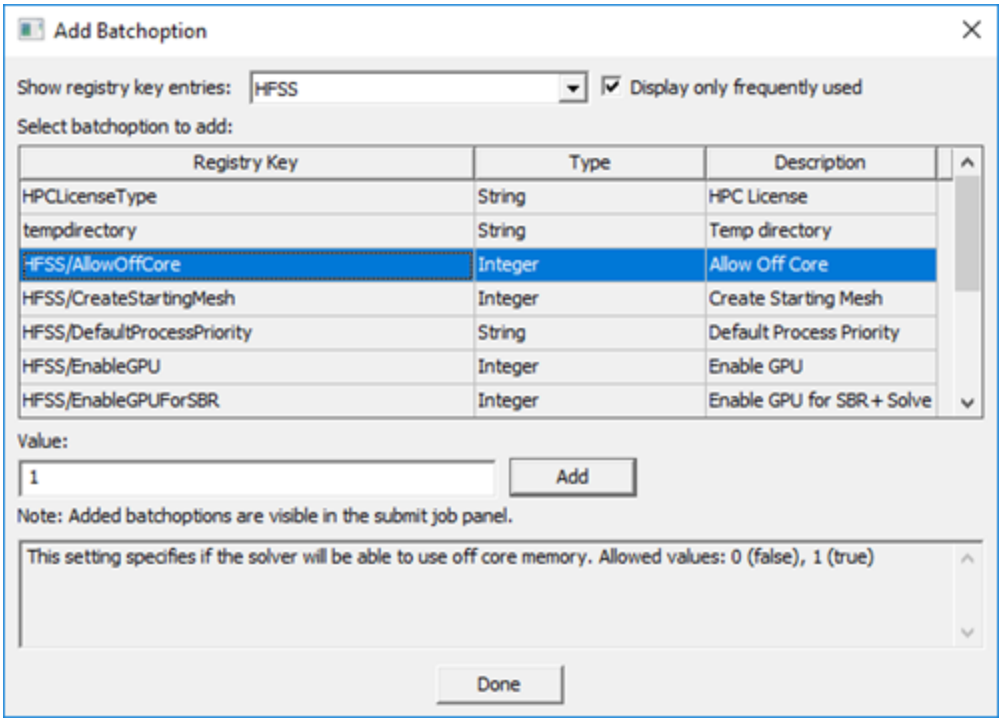
- Monitor job. You must enable this option to monitor the job from the user interface.
- Wait for license- whether to wait until a license if available before starting a simulation.
- Batch options. You can optionally specify -batchoptions in the text field. See detailed discussion of -batchoptions beginning under Running Maxwell from a Command Line.

The **Add...** button opens the *Add Batchoption* dialog box.

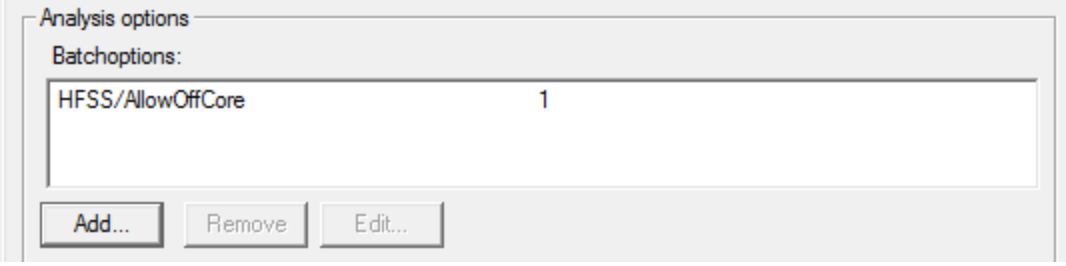


This dialog box provides access to all -batchoption commands. The drop-down menu lets you select specific categories, and you can choose to display only frequently used commands. You can edit and remove any batch options you specify.

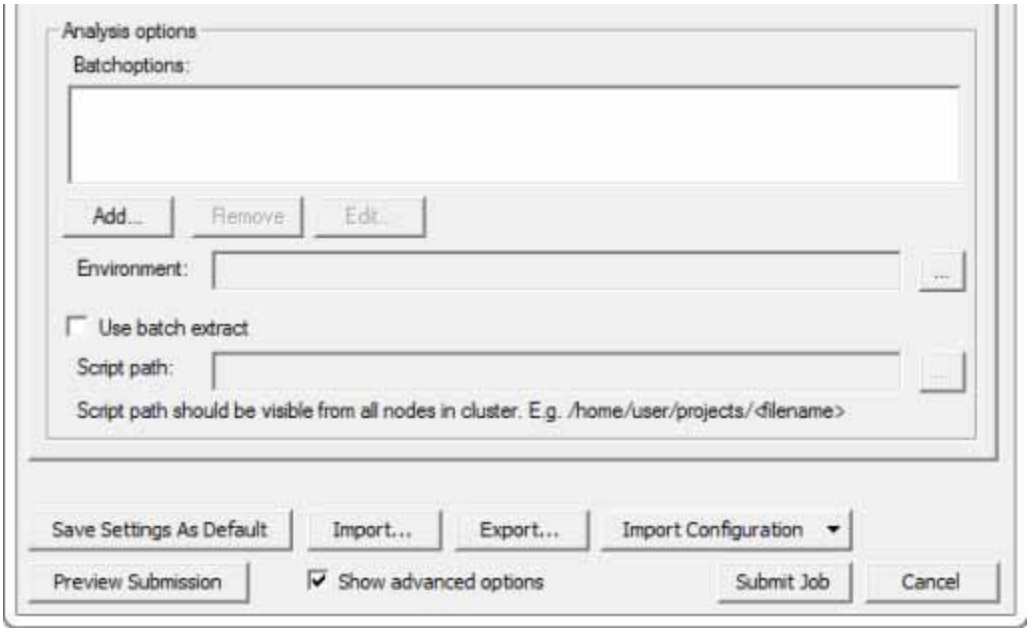
Select a Registry Key in order to show the current Value for the type. The lower field explains the meaning of the Type Value.



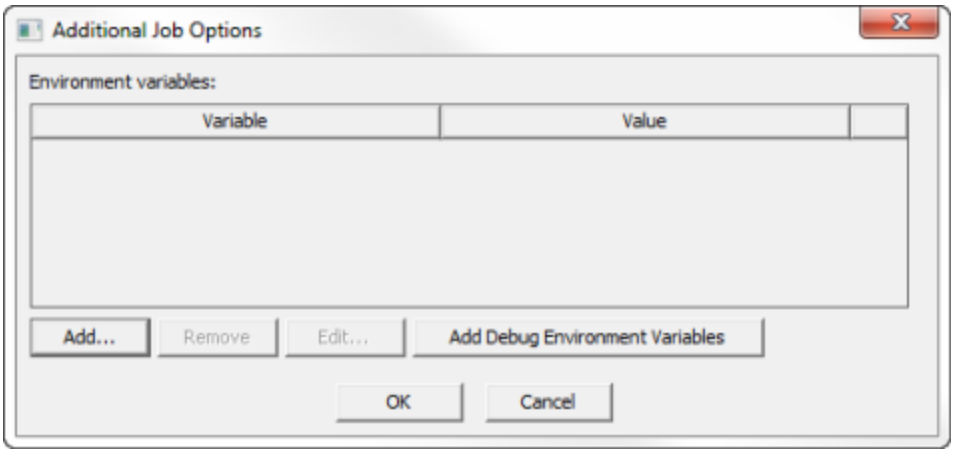
Any batchoptions for which you select **Add** will be visible in the *Submit Job* dialog box.



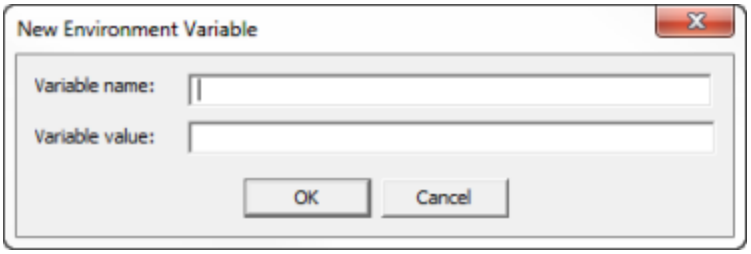
If you have the Show advanced options box checked in the *Submit Job* dialog box, the Environment field and the Use batch extract fields display.



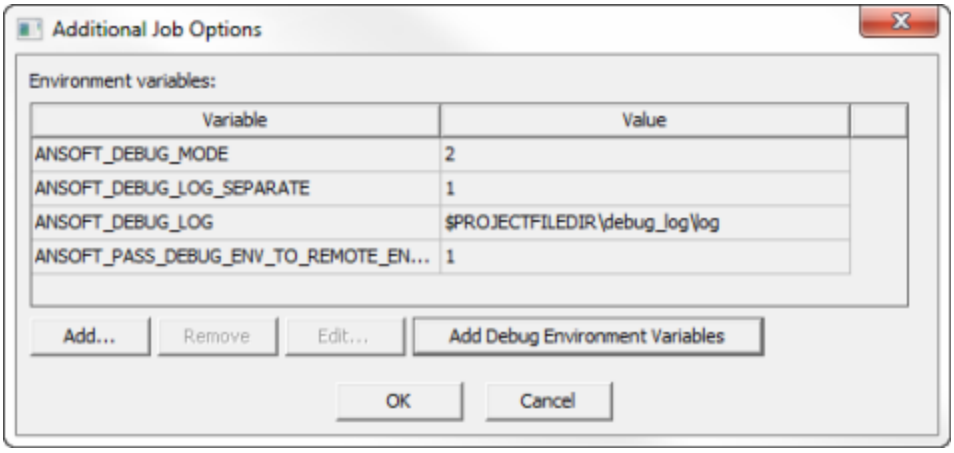
The Environment field lets you specify any Environment variables. Click the ellipsis button [...] to display the *Additional Job Options* dialog box.



Click the **Add...** button to open the *New Environment Variable* dialog box.



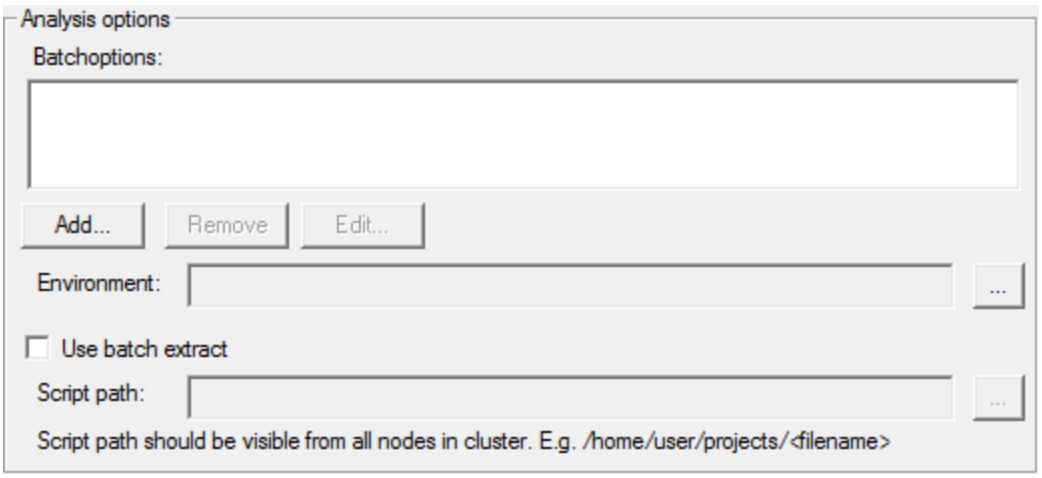
Here you can provide a Variable name and Variable value. Click OK to display the Variable in the Additional Job Options dialog box. Select a Variable to enables the **Remove** and **Edit...** buttons. You can also click **Add Debug Environment Variables**.



Any Variables that you add will be displayed in the Environment field of the *Submit Job* dialog box, if you have also enabled Show Advanced options.

Use Batch Extract for Windows HPC

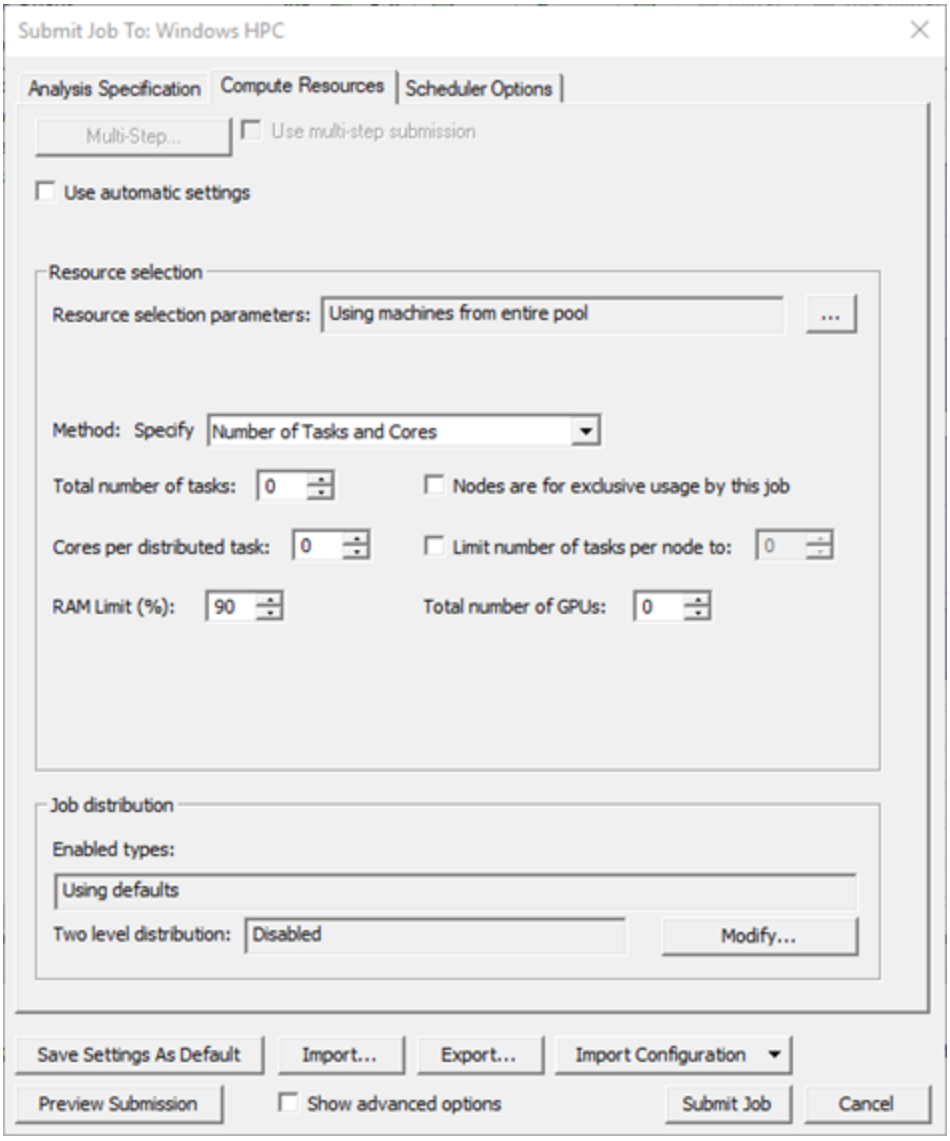
Selecting Show advanced options for Windows HPC also show the Use batch extract fields.



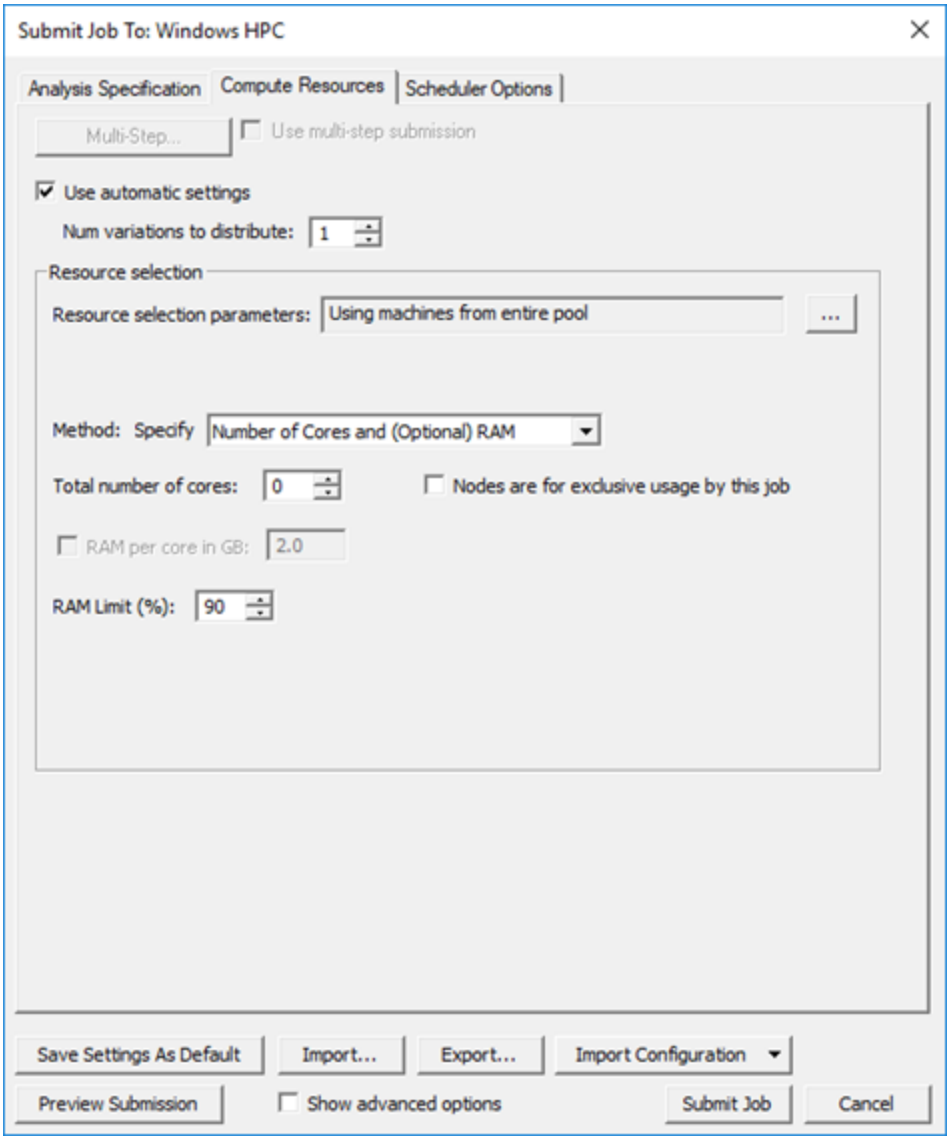
See the discussion on Running Maxwell from a Command line for a discussion of the solve information available through batch extract.

The **Preview Submission** button opens a window that shows the text commands that will be sent to the scheduler.

The following figure shows the *Compute Resources* tab of the *Submit Job To* dialog box.

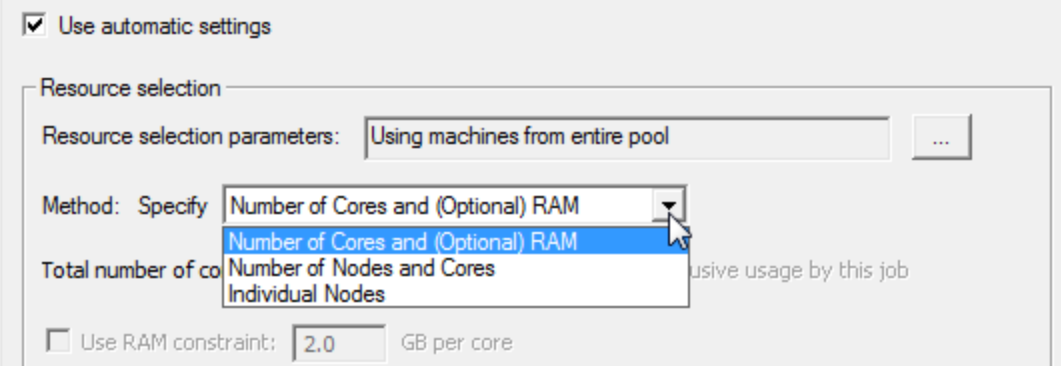


For Ansys Electronics Desktop configurations, the *Submit Job* dialog box includes a **Use automatic settings** check box that simplifies the *Compute Resources* tab.



HFSS, HFSS-3D Layout, and Icepak have **Use automatic settings** selected by default. With Use automatic settings selected, the Job distribution field is removed. For Use automatic settings, you can specify Resource selection parameters. The ellipsis button [...] opens the Compute Resource Selection Parameters dialog box. If you do not specify any parameters, the default is Using machines from the entire pool.

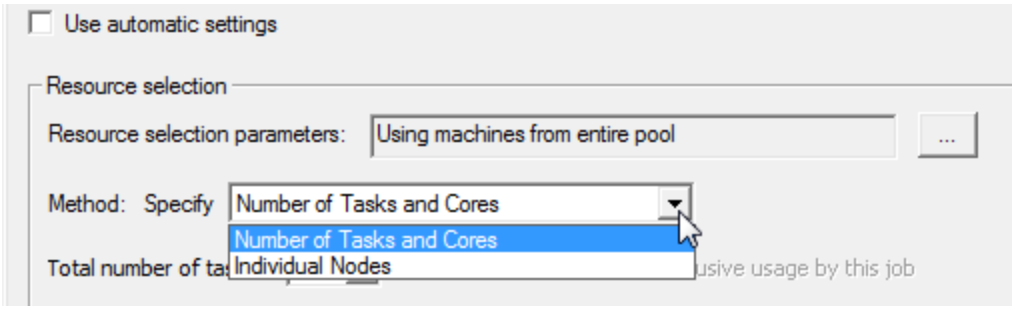
The Method field of the *Submit Job To* dialog box has a drop-down menu with two or three selections, depending on whether you select **Use automatic settings**.



Note:

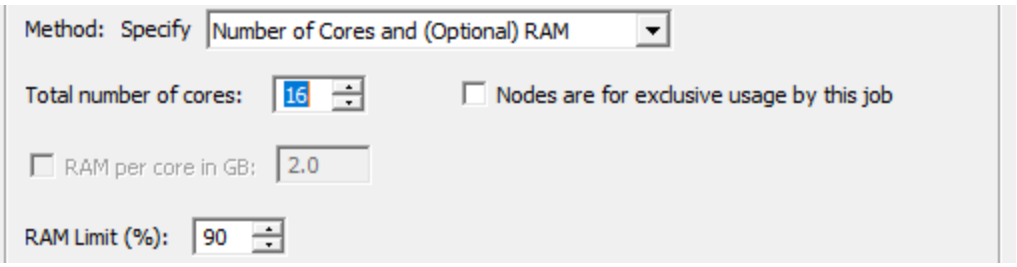
If you select Use automatic settings, Optimetrics variations will be solved sequentially. Other distribution types will be distributed automatically. It does distribute frequencies, domains, and use of multiple level domains.

If you uncheck or cannot access Use automatic settings, these two Methods are listed:

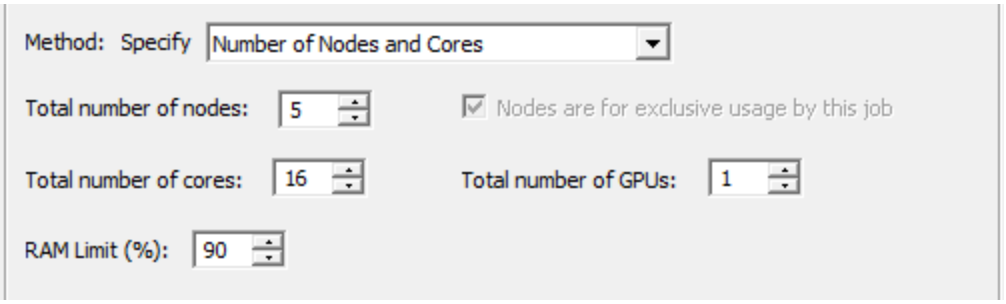


Each Method selection changes the available options listed:

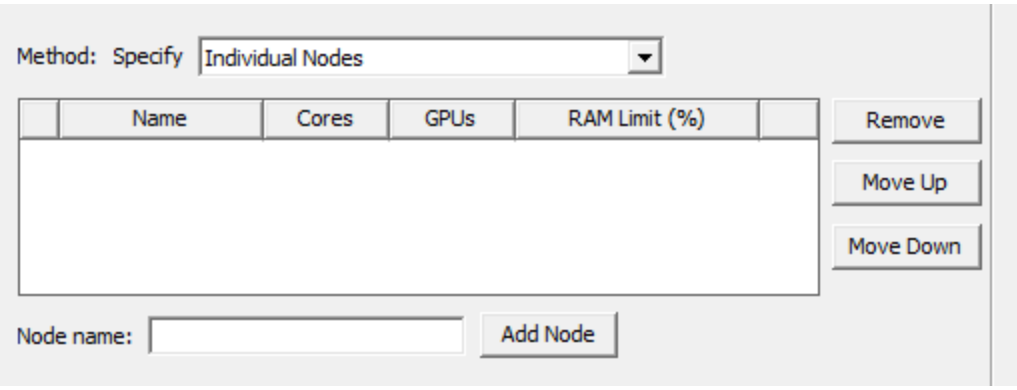
- Specify Number of Cores and (Optional) RAM



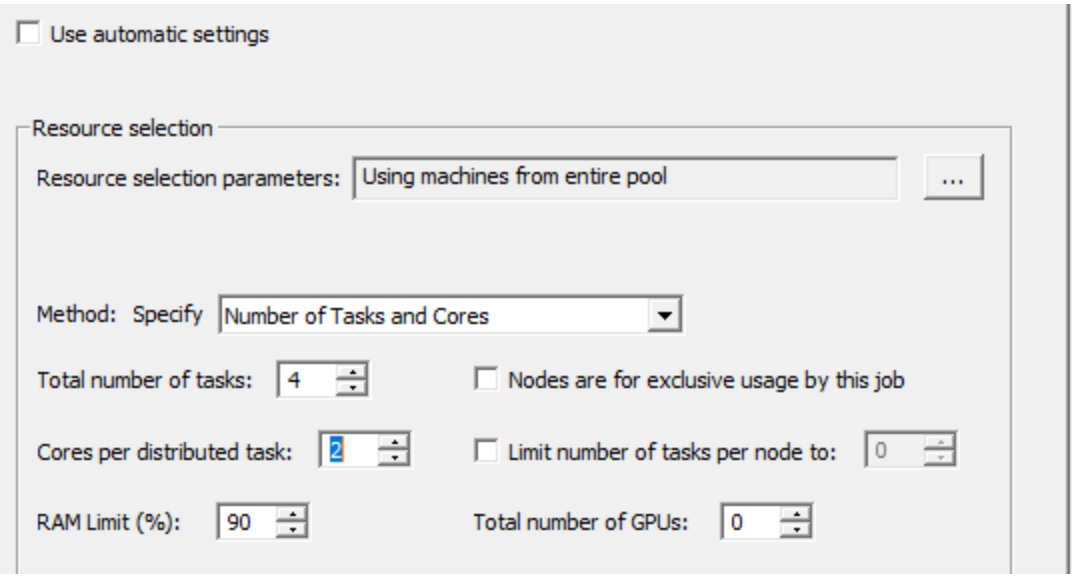
- Number of Nodes and Cores



- Individual Nodes



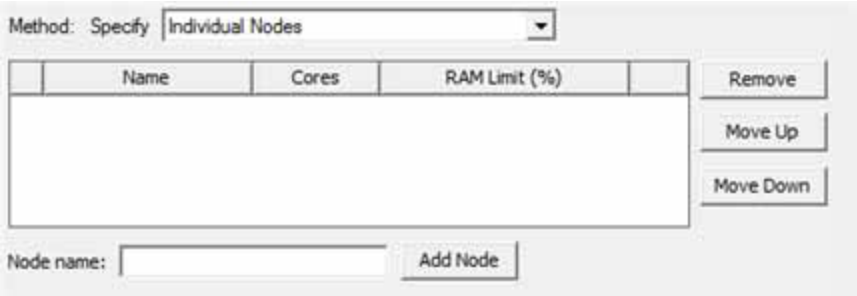
- Number of Tasks and Cores ("Use automatic settings" is unchecked for this option. Checking "Use automatic settings" means that you do not have to specify tasks or core parameters):



Individual Node List

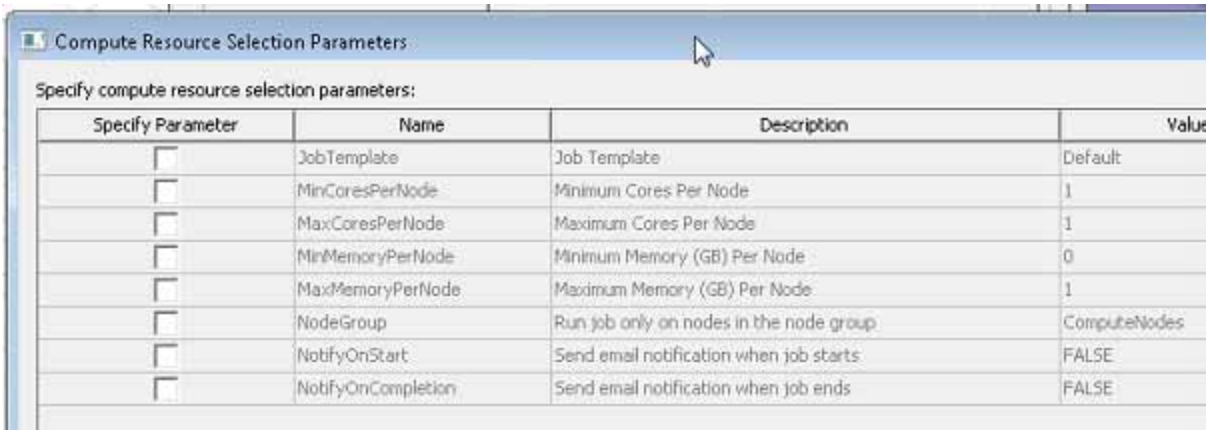
For Windows HPC jobs, you may either specify a node list, or specify job parallelization parameters, but not both.

If you select the Individual Nodes Method, you may specify a node list, and the Job parallelization controls are disabled. In this case, the node list should only include cluster nodes that are valid for the job. For each node, you enter the node name and add the node. In the table, you can specify the number of cores and the RAM limit as a percentage. You can use the Remove, Move Up and Move Down buttons to edit and order the list.



Compute Resource Selection Dialog

By default, you can draw from the entire pool. You can also click the ellipsis button [...] to open a *Compute Resource Selection* dialog box.



The resource selection parameters for Windows HPC jobs are:

- JobTemplate: Job Template - The JobTemplate may limit the job parameters or specify defaults values for job parameters
- MinCoresPerNode: Minimum Cores Per Node
- MaxCoresPerNode: Maximum Cores Per Node
- MinMemoryPerNode: Minimum Memory (GB) Per Node
- MaxMemoryPerNode: Minimum Memory (GB) Per Node
- NodeGroup: Run job only on nodes in the node group

- NotifyOnStart: If True, send email notification when job starts. Email notifications must be configured and enabled for the cluster by the administrator. (The cluster head node must run Windows HPC Server 2008 or above.)
- NotifyOnCompletion: If True, send email notification when job ends. Email notifications must be configured and enabled for the cluster by the administrator. (The cluster head node must run Windows HPC Server 2008 or above.)

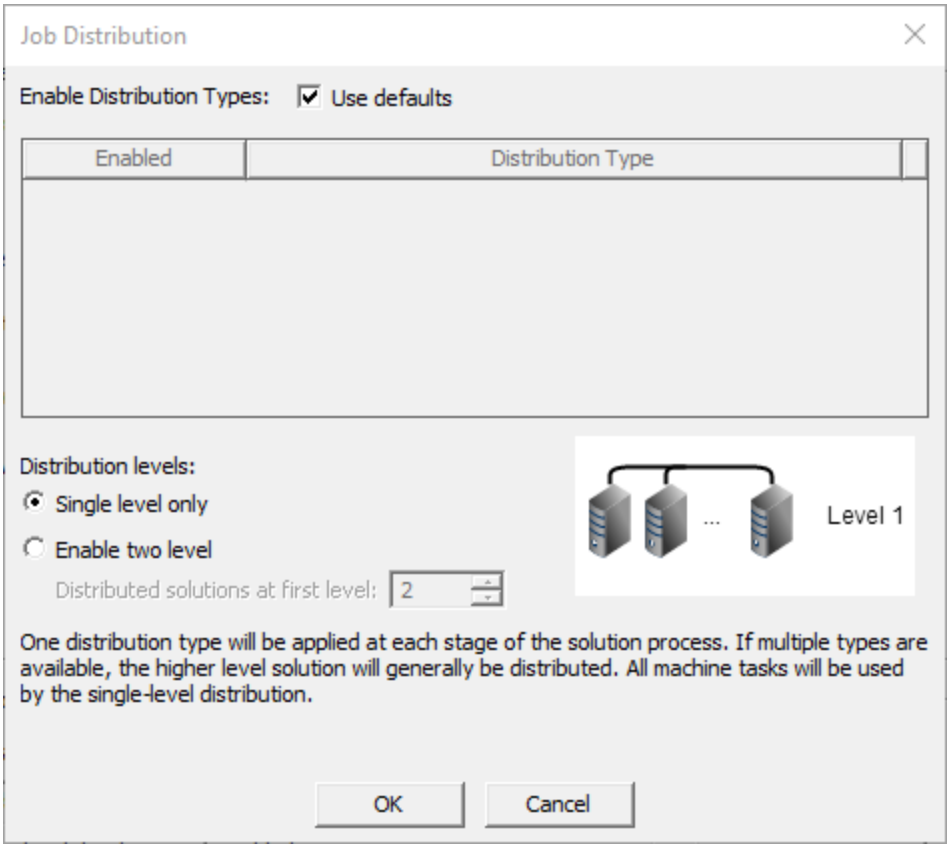
Job Parallelization

For Windows HPC jobs, you may either specify a node list, or specify the job parallelization parameters, but not both. The Job parallelization fields let you specify

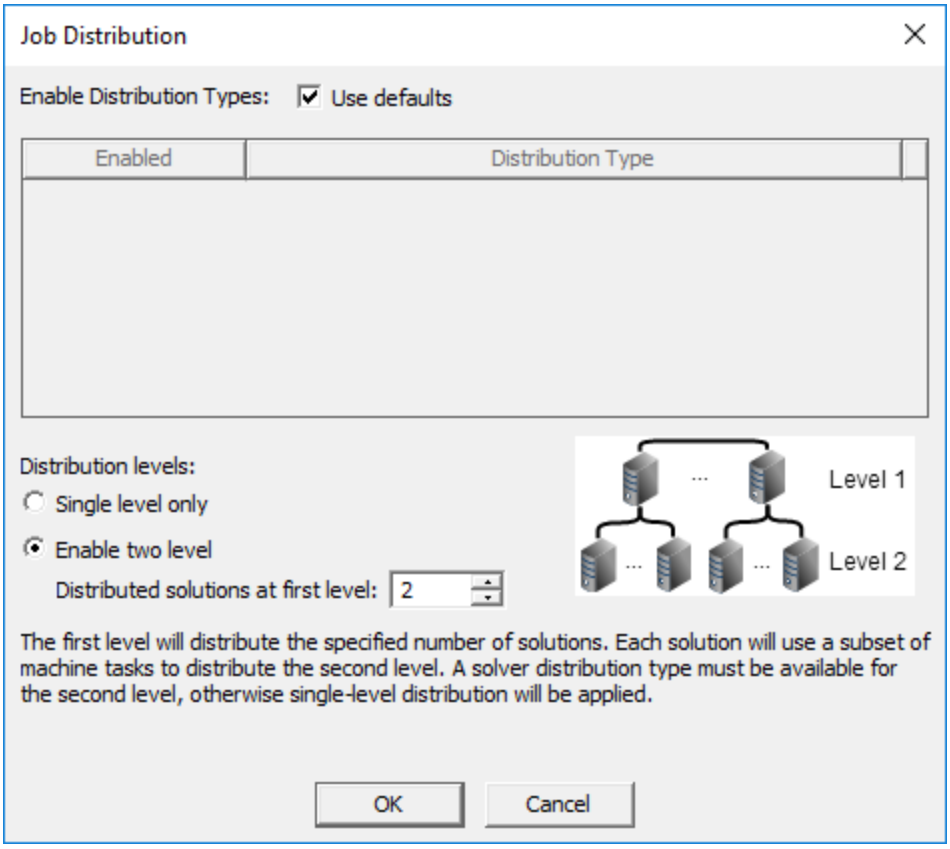
- Total number of tasks: The number of nodes requested for the job is the total number of tasks divided by limit on the number of tasks per node, rounded up if it is not an integer.
- Cores per distributed task. This determines the amount of multiprocessing per task.
- Whether nodes are for exclusive usage by this job
- Whether to limit the number of tasks per node to a value. If the "Limit number of tasks per node" check box is not checked, then the job is submitted with a job unit type of "Core".

Job Distribution

- Single level or two level distribution (*single level* is the default). Click **Modify** to display the *Job Distribution* dialog box and select the **Enable two level** option if applicable and desired.



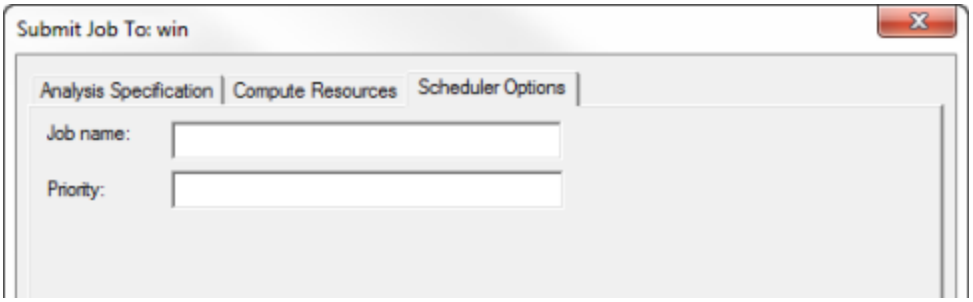
- Second level distribution operates within DSO. If available and enabled you can specify the number distributed solutions for level 1.



In response to a set of minimal constraints, the Scheduler may increase the resources assigned beyond the minimal values in order to meet the full set of requirements. For example, if you specify 7 distributed engines, with two processors per engine, and also limit the number of engines per node to 4, the scheduler may increase the number of cores used in order to meet the limit specified for engines per node. Notice that a preview of the Submit Job Results shows the number of resources assigned, and that the scheduler generated code includes an MPI specification.

Scheduler Options

The *Scheduler Options* tab provides for specifying the job name and/or the job priority. While the **Show advanced options** check box enables the display of Job submission options, no job submission options should be specified for Windows HPC.



Preview Submission

The Preview Submission button opens a window that shows a text description of the job to be submitted and the task used to start the product on one of the nodes.

The JOB PARAMETERS section contains information on parameter that apply to the job as a whole.

- The "Job resource parameters" section indicates whether the job has exclusive use of nodes, the job unit type, and the minimum and maximum number of units requested for the job, node group, and email notifications.
- The "Job attributes" section displays the job name and job priority.
- The "User Specified Compute Resource Attributes" displays the Resource selection settings.

The TASK PARAMETERS section contains information on parameters that apply to the Desktop task, which is the main task of the job.

- The "Desktop task resource parameters" section indicates the job unit type (which is the same as in the JOB PARAMETERS), and the minimum and maximum number of units requested for the Desktop task.
- The "Command Line section" displays the desktop task command line, including all arguments.
- The "Environment variables" section displays the environment variables that are set for the Desktop task; the same environment variables will also apply to all other tasks of the job.
- The "Working directory" section indicates the working directory in which the Desktop task will run.

Monitor Job

If you have checked the Monitor Job option on the *Submit Job To* dialog box, **AnalysisSpecification** tab, you can invoke the *Monitor Job* window by clicking **Tools> Job Management> Monitor Jobs...** This dialog box may also be brought up by checking the **Begin monitoring this job now** check box when a job is successfully submitting using the job submission dialog box. For more details, see Monitor Jobs window.

Related Topics

Submitting and Monitoring Ansys Electromagnetics HPC Jobs

Specifying the Number of Compute Resource Units for HPC Jobs

Windows® HPC Job Templates

The job templates are managed by the Windows HPC cluster administrator. Every cluster has at least one job template, the "Default" job template. Every job has an associated job template. If no job template is specified, then the "Default" job template is used. The job template controls two related aspects of the job submission process. When a job is submitted, there are a number of job parameters which may be specified. Each parameter has a set of valid values. For example, the Priority parameter has five valid values, Highest, AboveNormal, Normal, BelowNormal, and Lowest. The job template controls the default value of each parameter; this is the value that the parameter has if it is not specifically overridden by the submitter. For example, in the Default job template, the default value of the Priority parameter is Normal. The job template may also limit the allowed values of each parameter to a subset of the valid values. For example, a job template for privileged users could allow all five Priority values, which a job template for unprivileged users could limit the allowed Priority values to Normal, BelowNormal and Lowest.

Each job template is a Windows object with access controlled by an ACL (access control list). Instead of the usual "Read" or "Read & Execute" permissions, there is a "Submit Job" permission which corresponds to the right to submit a job with this job template. The cluster administrator may create job templates to limit or control access to cluster resources. For example, a job template with limited allowed job run times, or access to a limited set of compute nodes could be created by the cluster administrator. Specific users or user groups could be forced to use this limited job template by omitting access to the other job templates or by adding a deny access entry for the specified user or group to the other job templates. See the *Windows HPC Server 2008 Job Templates* white paper from Microsoft for additional details:

http://www.microsoft.com/en-us/download/confirmation.aspx?id=5659

Job templates may also be created to allow users to run jobs with limited knowledge of the appropriate job parameters. The cluster administrator creates a job template which has reasonable default values for the type of job to be run, and informs users which job template to use for each type of job. The template could also limit some parameters to only the subset of all values that are useful for the type of job associated with the template.

Related Topics

Integration With Microsoft Windows® HPC Scheduler

Windows® HPC Job Templates

Selecting Computation Resource Units (Job Unit Type)

Windows® HPC Job Credentials

High Performance Computing (HPC) Integration

Selecting Computation Resource Units (Job Unit Type)

The Job Unit Type is the smallest unit of processing resources used to schedule the job. This is one of the most important job properties. There are three options for the Job Unit Type: cores, nodes or sockets.

- **Cores**: Jobs are scheduled in units of cores, which may be also described as a CPU cores, logical processors, or CPUs. This is the smallest unit of granularity available. This selection allows the scheduler to start multiple tasks on a processor, if the total number of cores needed by the tasks is less than or equal to the number of cores on the processor. This selection may also allow the scheduler to distribute more of the computational load to processors with more cores than to processors with fewer cores.
- **Nodes**: Jobs are scheduled in units of nodes, hosts or machines. This is the coarsest level of granularity that may be selected. When this option is selected, only one task will run on any give node at any given time. This is useful in cases where it is not desirable to run multiple tasks on a single host. For example, if each task is multi-threaded, running multiple tasks on the same node may not be needed to fully utilize the computing resources on the node. This may also be preferred if the tasks are memory intensive, and multiple tasks would be competing for the limited memory resources.
- **Sockets**: A socket (which may also be called a NUMA node) is a collection of cores sharing a direct connection to memory. A socket will contain at least one core, and it may contain several cores. The socket concept may not necessarily correspond to a physical socket. Scheduling at the socket level may be useful in cases in which each task requires extensive use of the memory bus, and scheduling multiple tasks on the same socket would result in excessive bus contention.

Related Topics

Integration With Microsoft Windows® HPC Scheduler

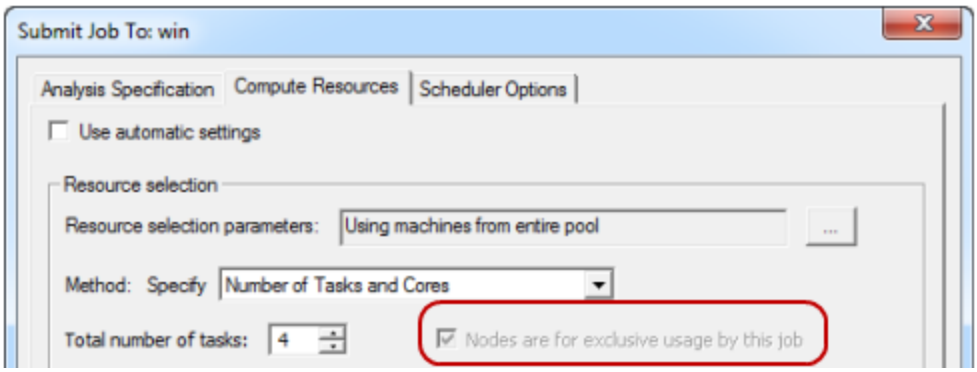
Windows® HPC Job Templates

Selecting Computation Resource Units (Job Unit Type)

Windows® HPC Job Credentials

Windows® HPC Non-Exclusive Jobs

In previous releases, when submitting jobs to a Windows HPC cluster using the **Submit Job To** dialog box, the exclusive node usage property of the job is on the **Compute Resources** tab always set to “true”.



For many cases, this requirement prevents problems. One type of problem that can occur is related to processor affinity. In general, EBU jobs manage processor affinity on their own, and request a single core for each Windows HPC task. If the cluster does set the affinity for the job, then the EBU job cannot use all of the resources allocated to the job because the cluster sets the affinity to allow only a single processor for each Windows HPC task.

Recent versions of Windows HPC allow three choices for the Cluster AffinityType setting:

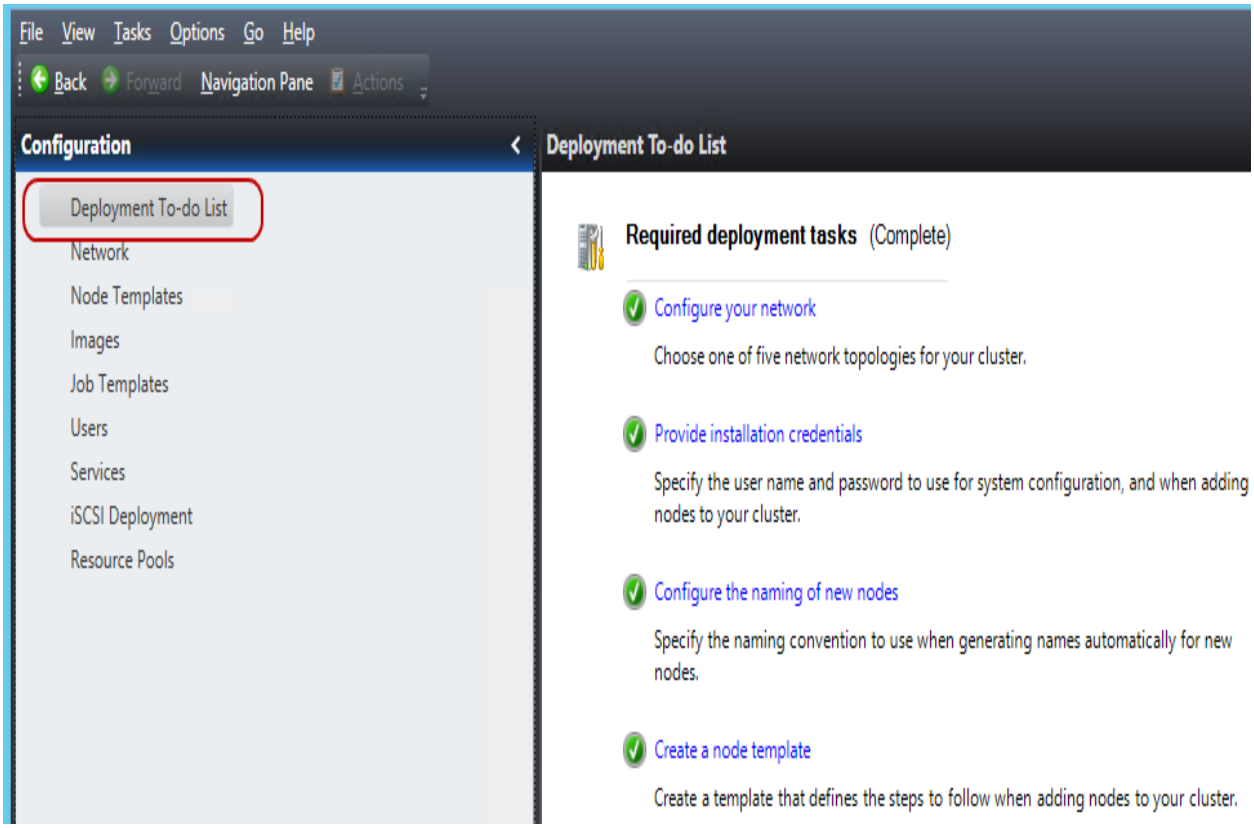
- No Jobs: Affinity is not set for any jobs
- Non Exclusive Jobs (default): Affinity is not set for exclusive jobs
- All Jobs: Affinity is set for all jobs

If the AffinityType is set to “No Jobs”, then we can allow non-exclusive jobs in many cases, but non-exclusive jobs will be disallowed if you specify an explicit host list (because all resources on all specified hosts are used), if you specify nodes and cores (to ensure that the job has exclusive access to the specified number of nodes), or if you specify multi-threading (to prevent a single task from being split across hosts). If the AffinityType is set to “Non Exclusive Jobs”, then exclusive jobs will be disallowed for all cases, the same as the restrictions for the previous release.

The previous job submission dialog always has the “exclusive” checkbox disabled for Windows HPC jobs, so that “exclusive” jobs cannot be submitted to a Windows HPC cluster. The “exclusive” checkbox is now enabled for Windows HPC jobs. This allows you to submit exclusive jobs to a Windows HPC cluster. If this checkbox is checked, but the submission parameters and Windows HPC cluster AffinityType do not allow submission of an exclusive job, then an error message is displayed when you submit or preview the job.

Setting the Windows HPC Cluster Affinity Type

A Windows HPC Cluster administrator is able to modify the Affinity Type setting for the cluster. The administrator should start the Windows HPC Cluster Manager. The first step is to select the Configuration button on the lower left pane, then select the Deployment To-do List in the upper left pane. Then, the upper right pane displays the Deployment To-do List, as shown below.



Next, the "Configure job scheduler policies and settings" should be selected. This is under the heading "Optional deployment tasks".

Optional deployment tasks

Validate your environment before deploying nodes

Run a set of short diagnostic tests to find common problems that can affect node deployment.

Go online to get the latest set of tests

Add an operating system image

Create a new image or load an existing image to use with your node templates when deploying nodes.

Add or remove users

Add or remove users or administrators for your cluster.

Add nodes to your cluster

Deploy nodes, import a node XML file, or add preconfigured nodes to your cluster.

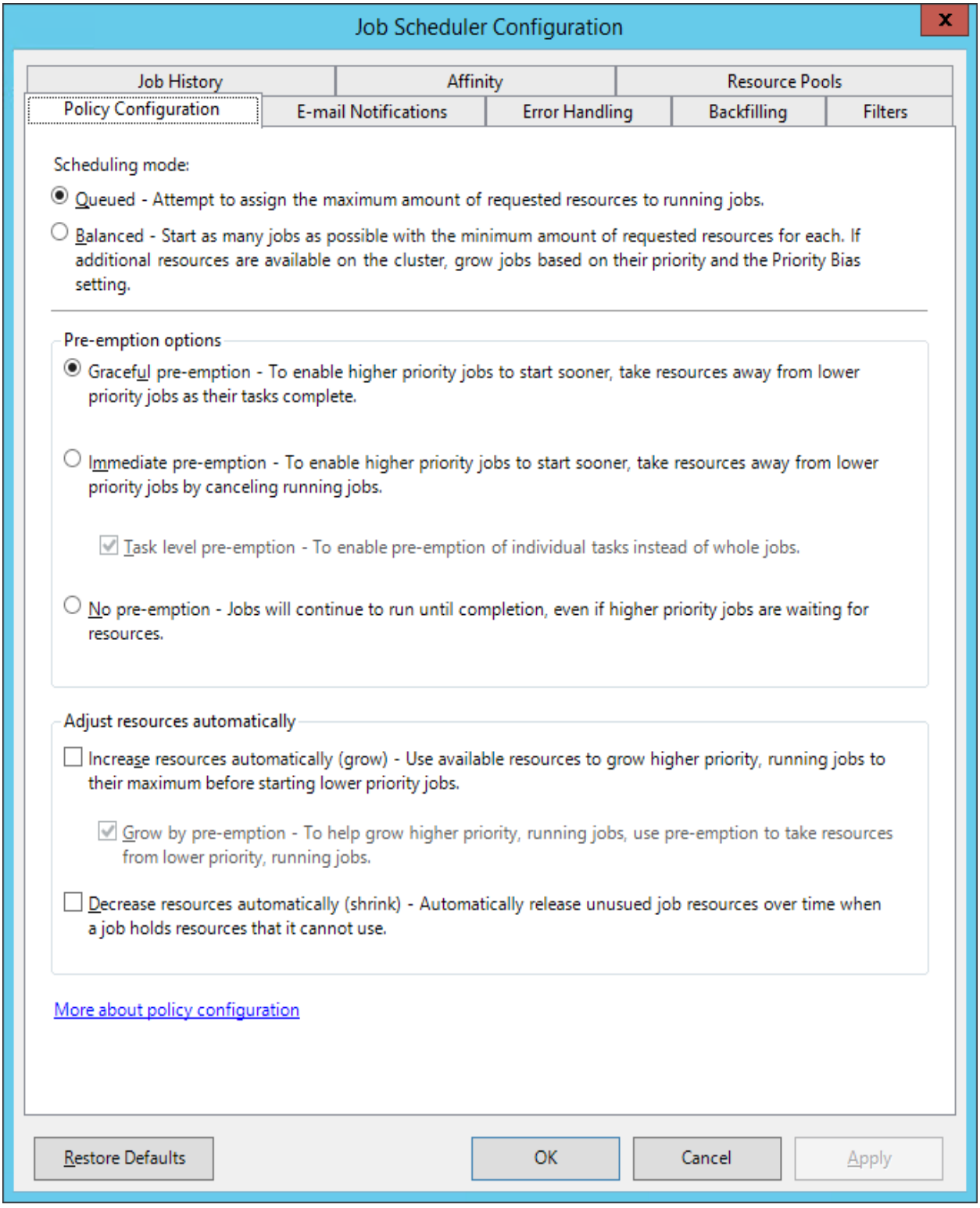
Manage drivers

Add device drivers to the operating system images.

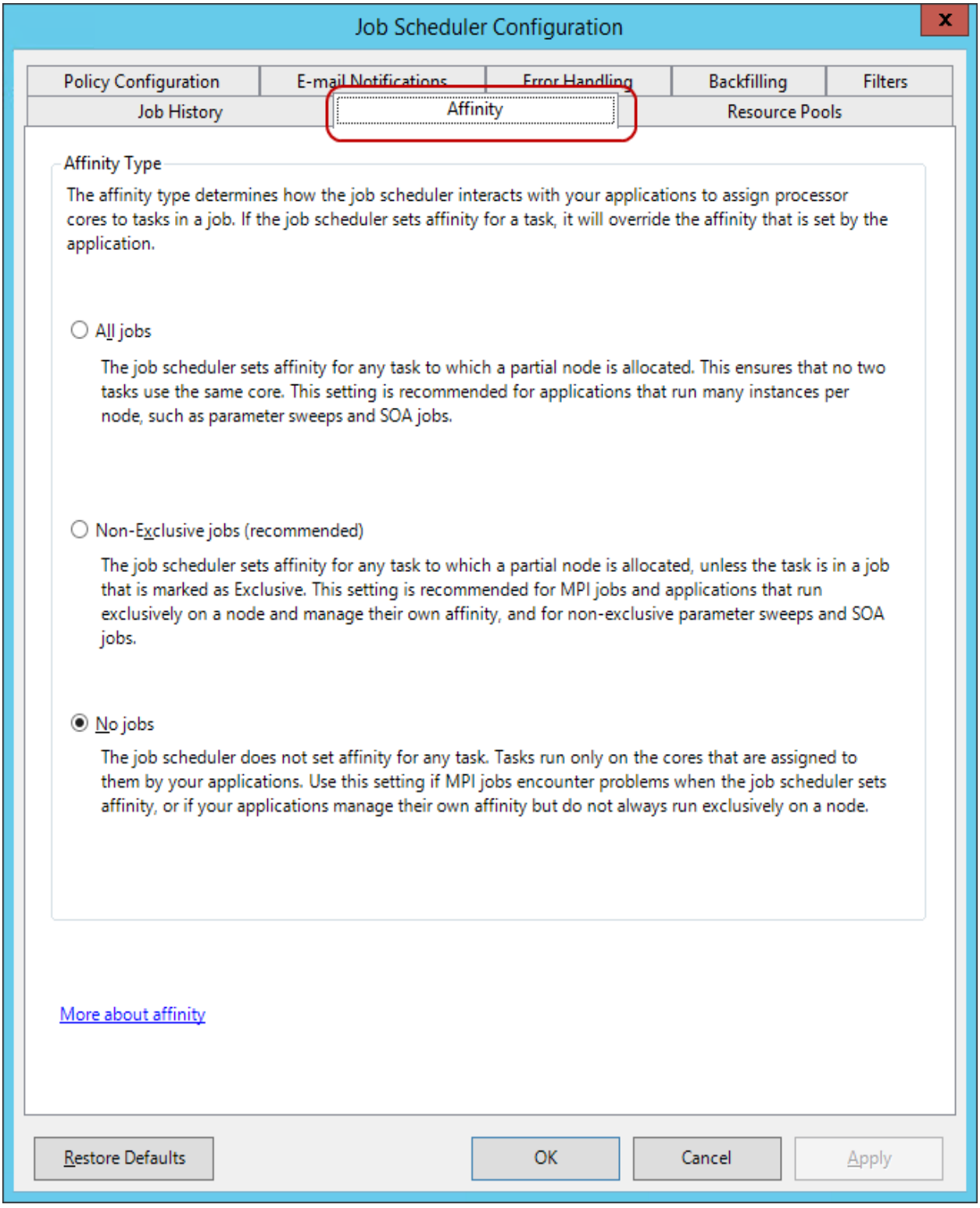
Configure job scheduler policies and settings

Customize policies, error handling, filters, and e-mail notifications for your cluster.

The "Job Scheduler Configuration" dialog pops up, as shown below.



Next, select the "Affinity" tab should be selected.



The cluster administrator may now select a radio button to select the desired Affinity Type:

- All jobs
- Non-Exclusive jobs
- No jobs

In order to submit non-exclusive jobs, the "No jobs" option should be selected. The "Non-Exclusive jobs" option may be selected, but this will not allow non-exclusive Ansys Electromagnetics jobs to be submitted to the cluster. The "All jobs" option is not supported for Ansys Electromagnetics jobs. The OK or Apply buttons may be used to apply the selected option to the cluster.

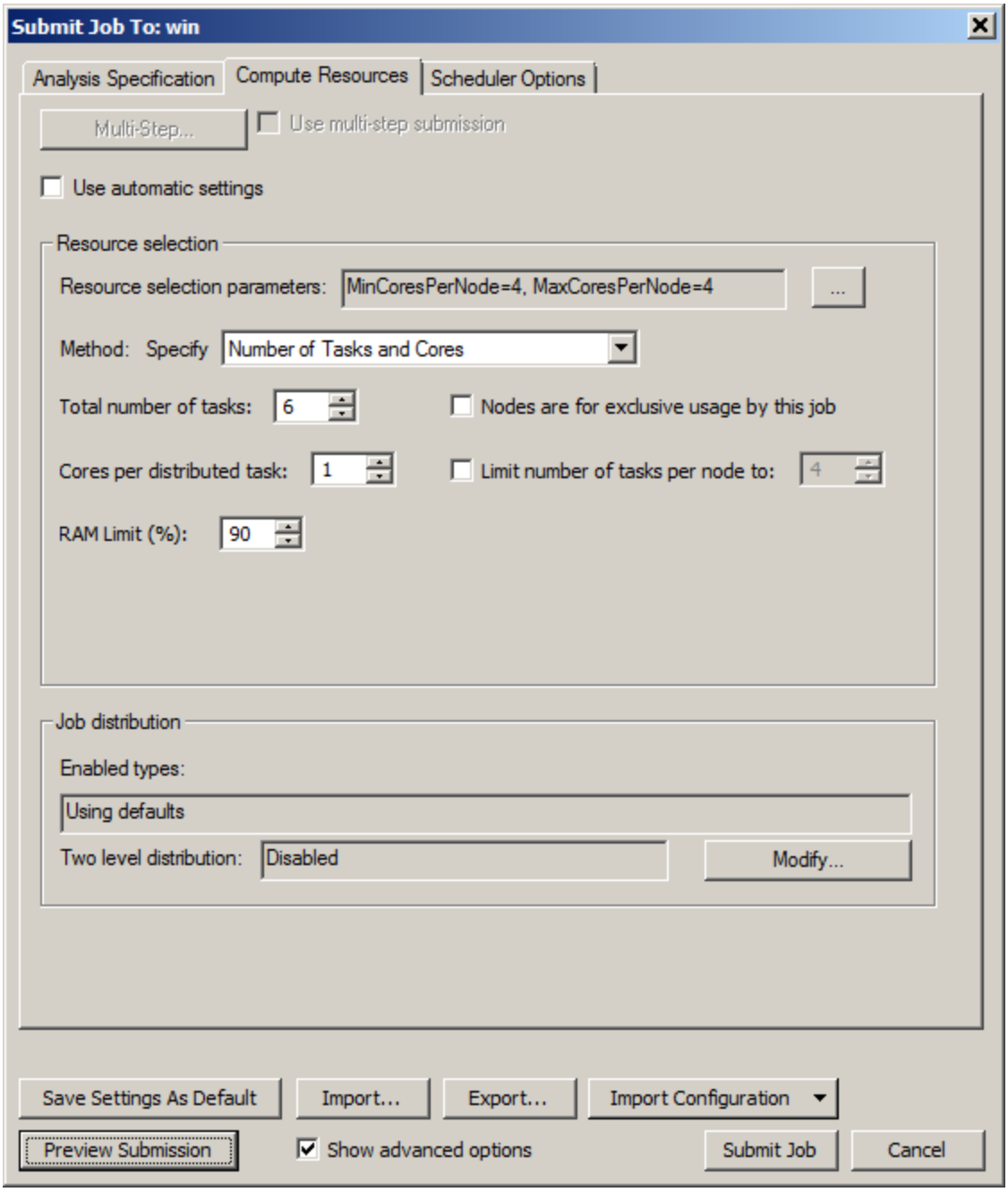
Examples with "Submissions with Nodes are for exclusive usage by this job" Unchecked and Checked

Most of the following screen shots are for the same project file submitted to a Windows HPC cluster with the AffinityType set to "No Jobs". The scheduler resource selection parameters are the same for all of these screen shots:

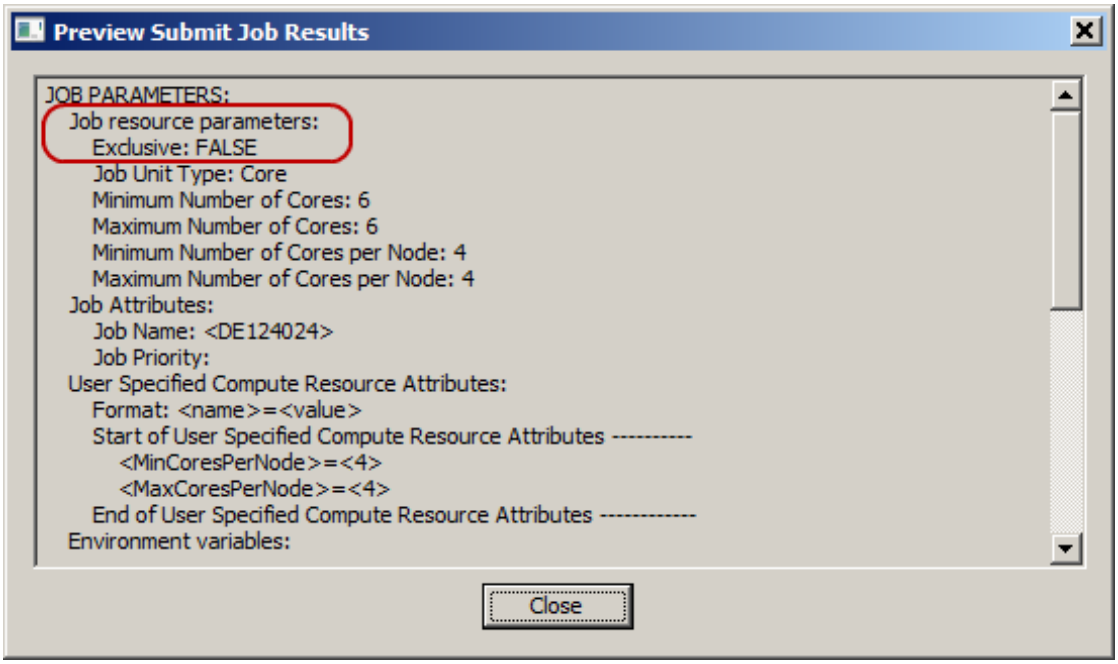
- MinCoresPerNode=4
- MaxCoresPerNode=4

The first set of screen shots show submission of a job to a Windows HPC cluster in which the "Use automatic settings" checkbox is not checked, and the resource selection method is "Specify Number of Tasks and Cores". The exclusive setting for the job is controlled by the "Nodes are for exclusive usage by this job" checkbox. In previous releases, this checkbox was always disabled and checked for Windows HPC jobs. Now, it is enabled, and the user may select whether it is checked or unchecked. If it is checked, then the behavior is the same as previous releases. If it is unchecked, then the job is submitted with the exclusive parameter for the job set to false.

The following screen shot shows this configuration with a Submit Job with Task and Cores specified and nodes as non-exclusive



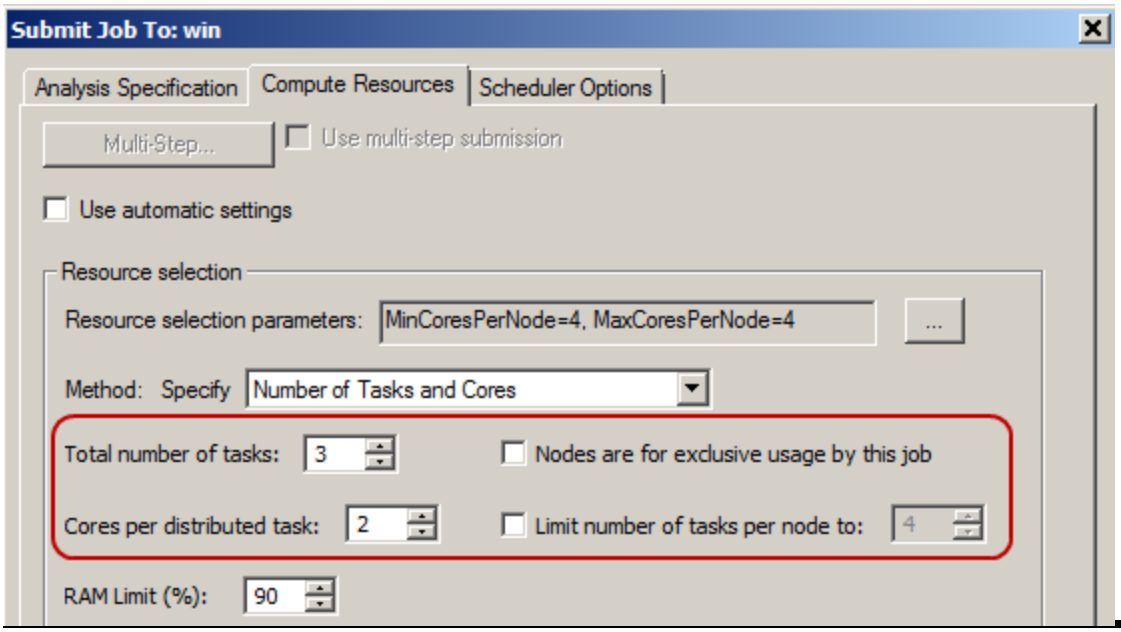
The next figure shows Preview for the job submission indicating that the exclusive parameter for the job is false.



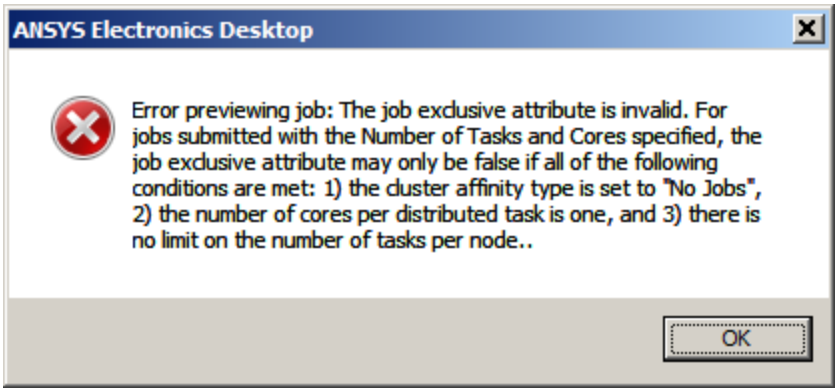
Jobs may only be submitted using this resource selection method with the exclusive parameter set to false if all three of the following conditions are met:

1. The cluster AffinityType is "No Jobs",
2. There is no multiprocessing (i.e., the number of cores per task is one), and
3. There is no limit on the number of tasks per node.

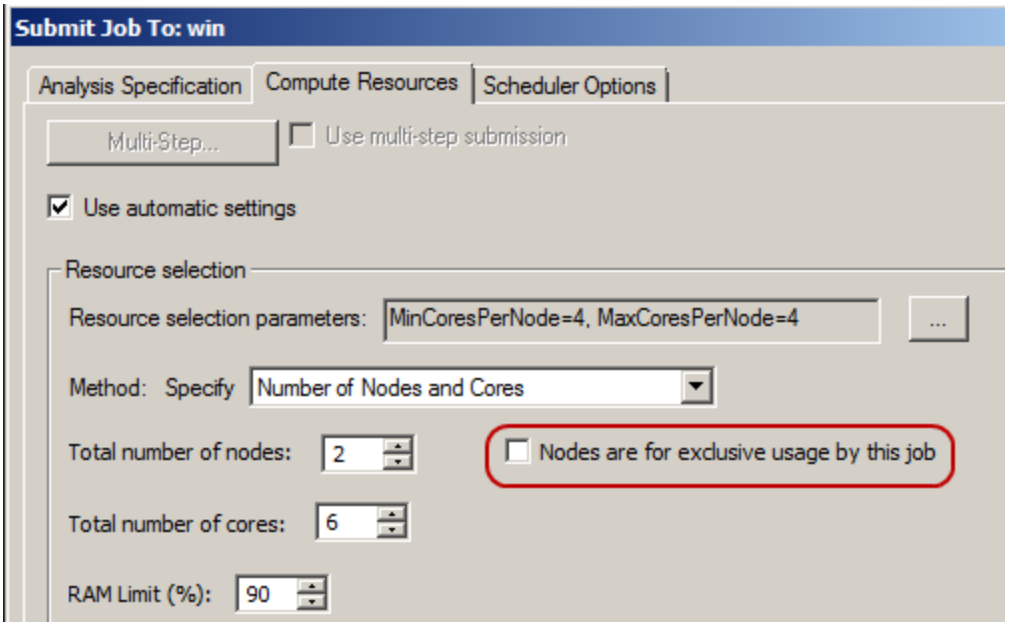
If any of these requirements are not met, then the job is not submitted or previewed, and an error message appears. The following figure shows this configuration.



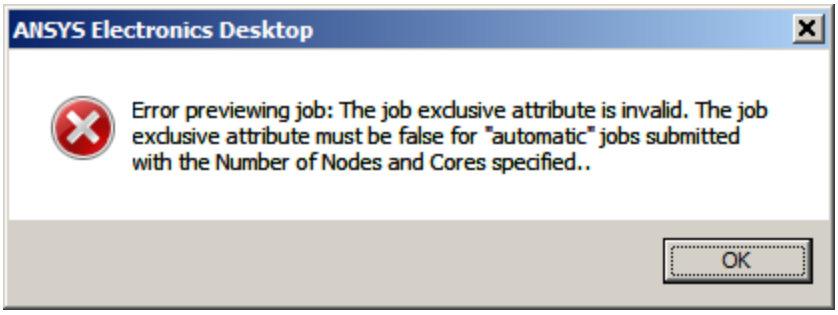
The following figure and screen shot ErrorMsgTasksAndCoresNonExMP.png shows the error message box shown to the user.



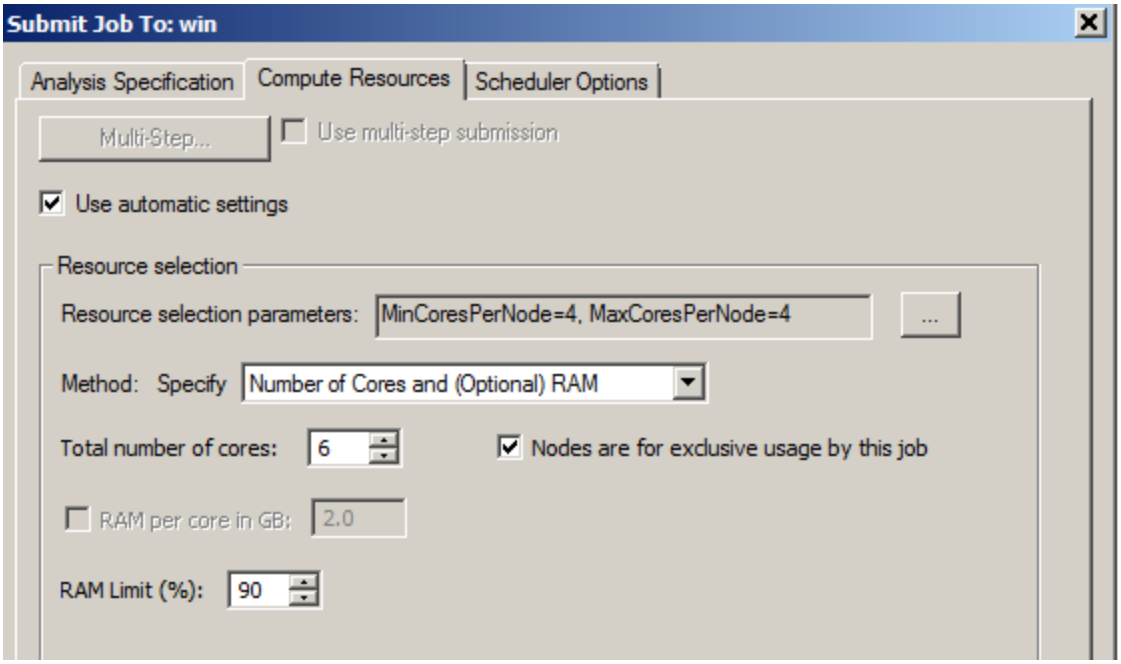
The next example shows submission of a job to a Windows HPC cluster in which the "Use automatic settings" checkbox is checked, and the resource selection method is "Specify Number of Nodes and Cores". The exclusive setting for the job is controlled by the "Nodes are for exclusive usage by this job" checkbox. In previous releases, this checkbox was always disabled and checked for Windows HPC jobs. Now, it is enabled, and the user may select whether it is checked or unchecked. If it is checked, then the behavior is the same as previous releases. If it is unchecked, then the job is not submitted or previewed because only exclusive jobs are allowed when using this resource selection method, and an error message box is shown to the user in this case. The following figure shows this configuration:



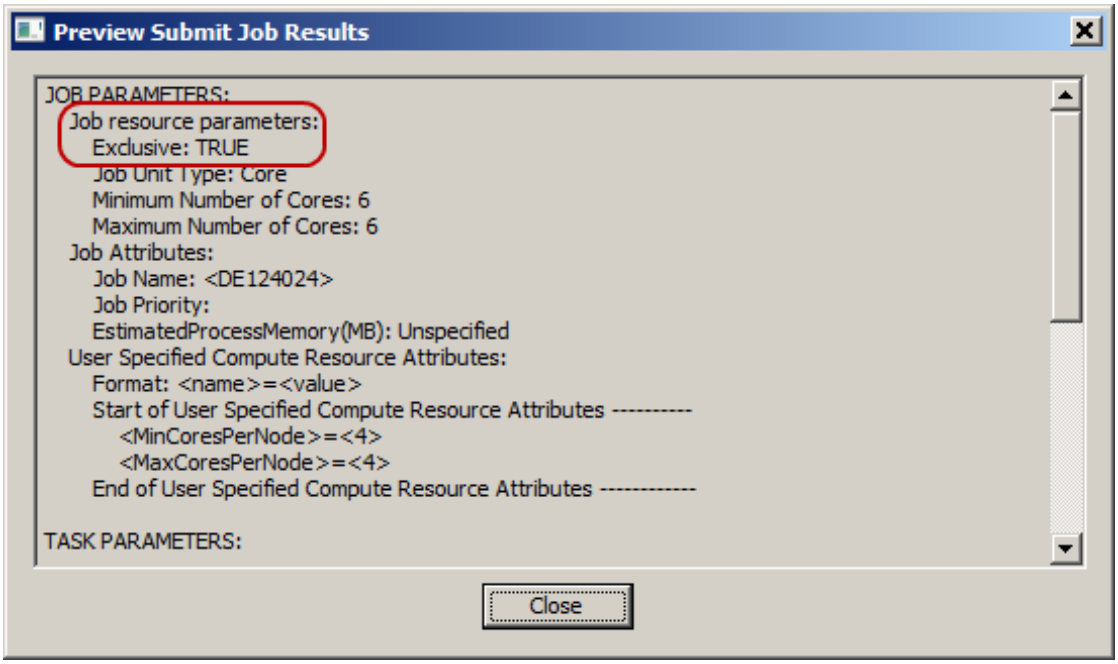
The following figure shows the error message.



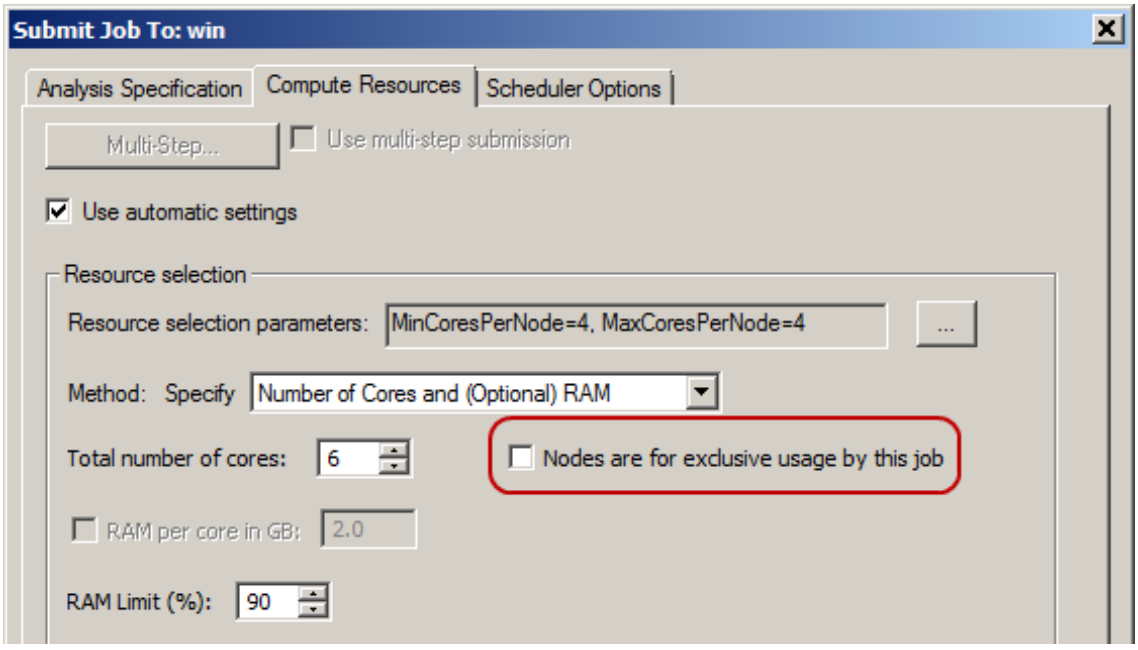
The next example show submission of a job to a Windows HPC cluster in which the "Use automatic settings" checkbox is checked, and the resource selection method is "Specify Number of Cores and (Optional) RAM". The exclusive setting for the job is controlled by the "Nodes are for exclusive usage by this job" checkbox. In previous releases, this checkbox was always disabled and checked for Windows HPC jobs. Now, it is enabled, and you may select whether it is checked or unchecked. If it is checked, then the behavior is the same as previous releases. The following figure shows this configuration.



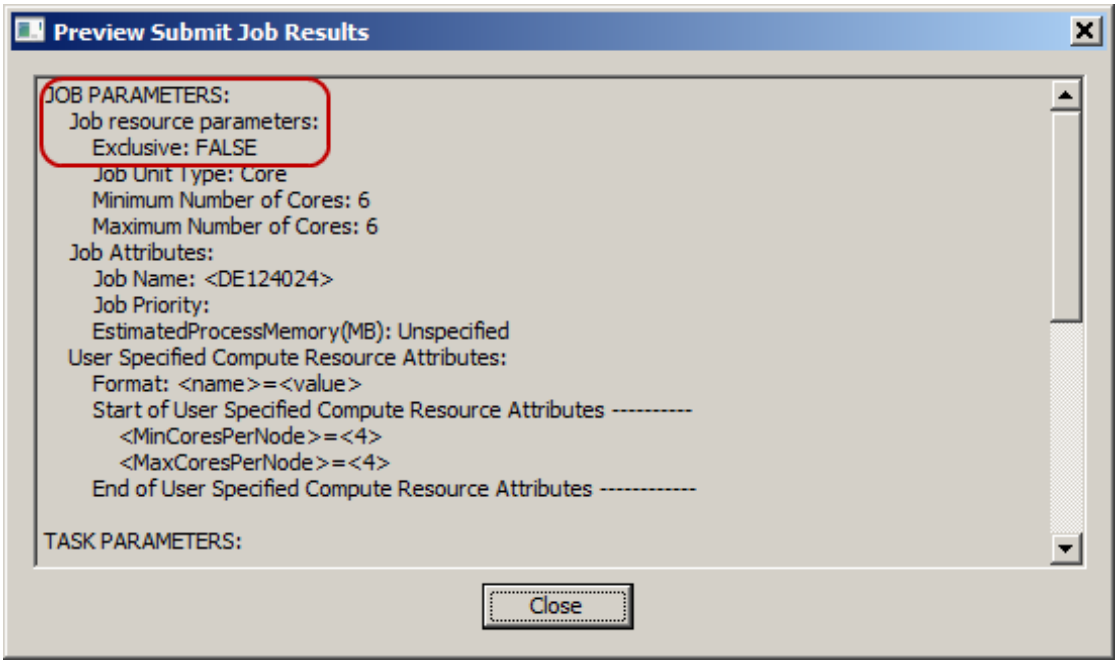
The following figure shows the Job Preview for this case, indicating that the exclusive parameter for the job is true.



If the exclusive checkbox is unchecked, then the job is submitted with the exclusive parameter set to false. The following figure shows this configuration.

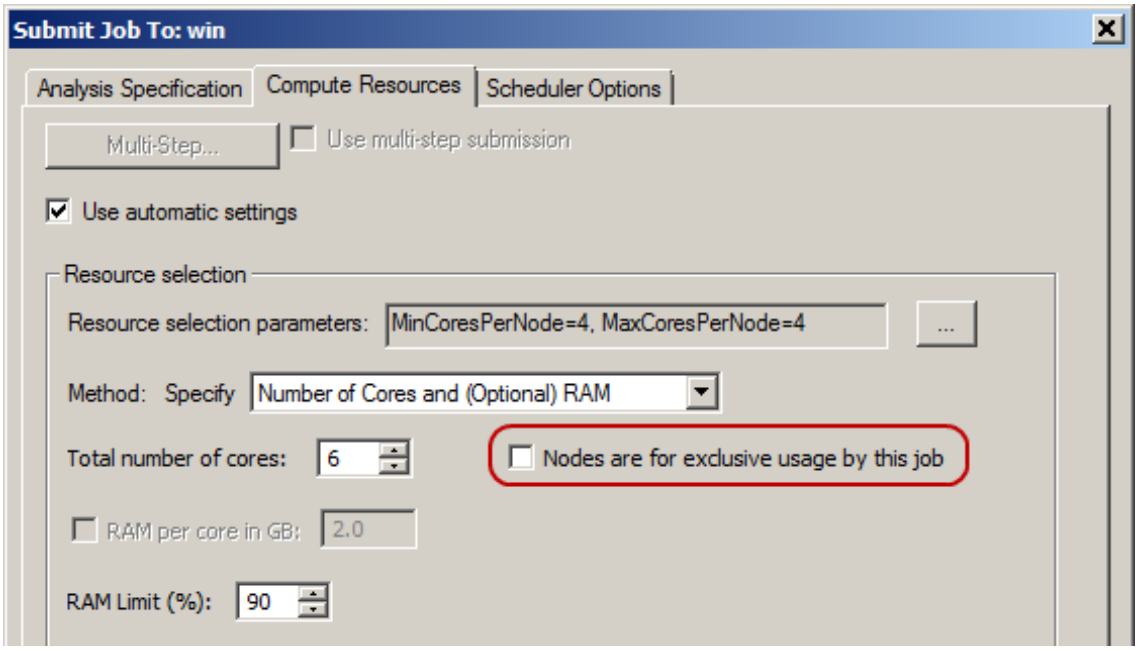


The following figurescreen shot PreviewCoresAndRAMNonEx.png shows the Job Preview for this case, indicating that the exclusive parameter for the job is false.

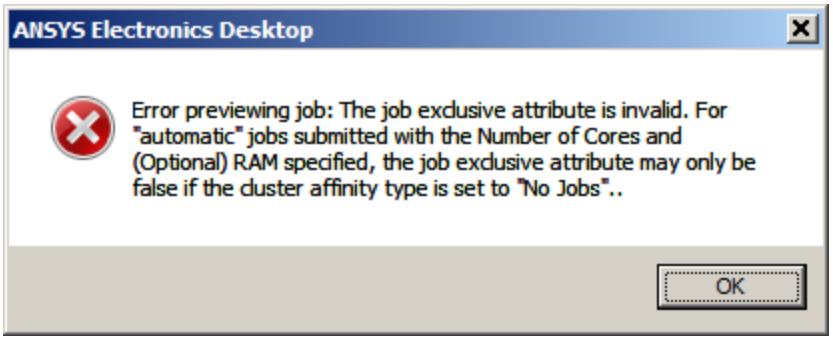


The last example resembles the previous one, except that the Windows HPC cluster AffinityType is set to "Non Exclusive Jobs" in this case. The job is submitted with the "Use automatic settings" checkbox in the checked state, and the resource selection method set to "Specify Number of Cores and (Optional) RAM". Because the AffinityType is not "No Jobs", only exclusive jobs may be submitted to this cluster. If the exclusive checkbox is not checked, then the job is not submitted or previewed, and an error message box is displayed to the user. The following figure shows this

configuration.



The following figure shows the error message.



Related Topics

Integration With Microsoft Windows® HPC Scheduler
Windows® HPC Job Templates
Selecting Computation Resource Units (Job Unit Type)
High Performance Computing (HPC) Integration

Windows® HPC Job Credentials

Normally, a user will be prompted for the credentials used to submit a job. One way to simplify this process is to use the "cluscfg setcreds" command to set the user's credentials in the credentials

cache. If this is done, then no password needs to be supplied for a job submitted for the specified user. Here is a cluscfg command that may be used to set the user credentials in the credentials cache:

```
cluscfg setcreds /password:* /scheduler:cluster_name
/user:domain\user_name
```

Here:

- cluster_name = the name of the cluster (hostname of the head node)
- domain = optional domain name; if omitted, the following \ should also be omitted
- user = user name

When this form of the command is used, the user is prompted for the password and also asked if the password should be remembered (cached).

See the following web page for more information on the cluscfg setcreds command:

http://technet.microsoft.com/en-us/library/cc947669(WS.10).aspx

Related Topics

Integration With Microsoft Windows® HPC Scheduler

Windows® HPC Job Templates

Selecting Computation Resource Units (Job Unit Type)

Windows® HPC Job Credentials

High Performance Computing (HPC) Integration

Integration with Grid Engine (GE)

Before Sun was acquired by Oracle, this job scheduler was an open source product, and it was known as Sun Grid Engine (GE). Since the acquisition, the product has been renamed Oracle Grid Engine (OGE), and new versions are expected to be closed source versions. The Open Grid Scheduler project hosted on SourceForge plans to continue maintaining an open source version. Univa has taken over support of Oracle Grid Engine, and it is now called Univa Grid Engine. Any of the versions may also be called **Grid Engine** (**GE**).

The Grid Engine scheduler is only supported on Linux. With GE, jobs may be submitted in any of the following ways:

- Using GE commands (qsub, etc.) or the GE gui (qmon)
- Using the generic scheduler GUI in local mode
- Using the generic scheduler GUI in service mode

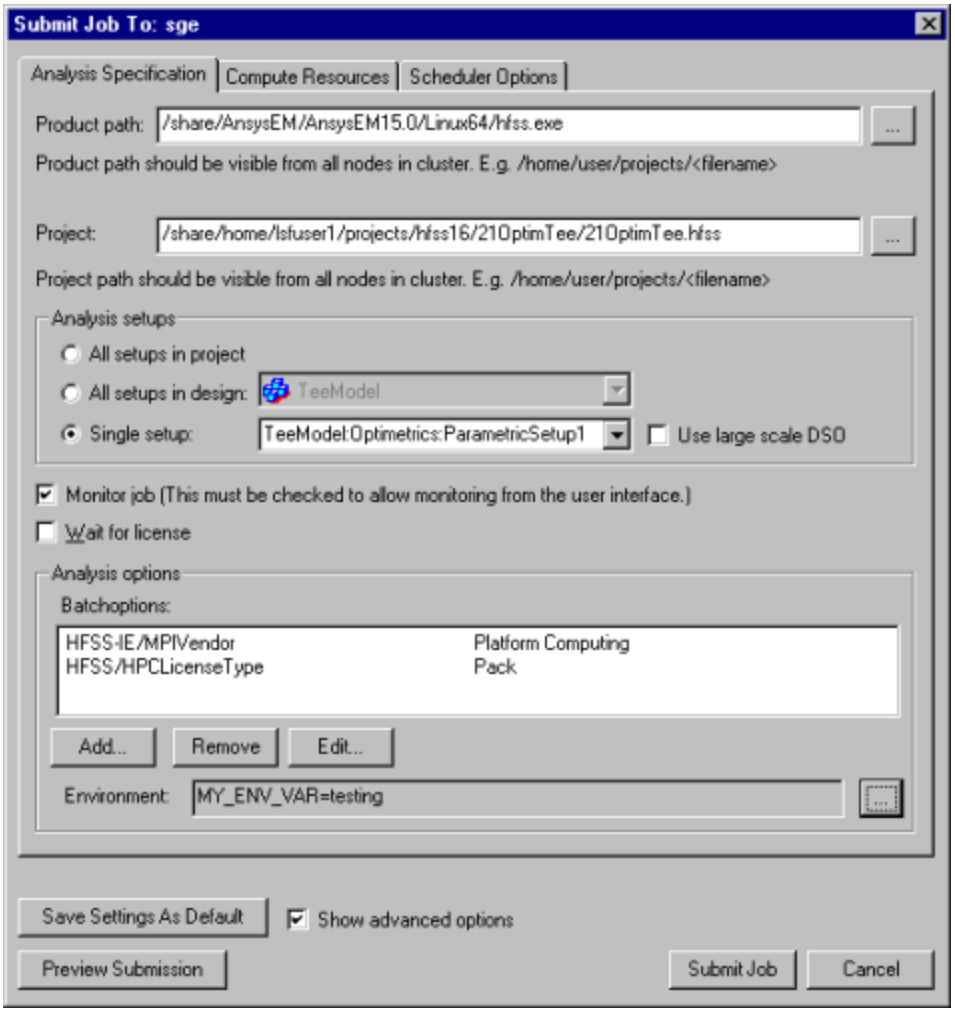
This release has been tested with the following versions of GE:

- SGE (Sun) 6.2u6
- OGS (Oracle) 2011.11
- Univa: 8.1.3

Ansys Electromagnetics products support Grid Engine (GE) for Serial analysis, Multi Processing and Distributed Analysis. Models with parametric sweeps can use Large Scale DSO. With GE, the

Ansys Electromagnetics job doesn't require graphics. Ansys Electromagnetics job progress can be monitored through GE(SGE) commands or in the dialog opened through **Tools>Job Management>Monitor Jobs...**.

In addition to the command line interface, you can also use a Job Management user interface to submit jobs.



Related Topics

Job Management User Interface for GE

Command Line Enhancements for Ansys Electromagnetics Desktop Products

Installation of Ansys Electromagnetics Tools on GE

Windows:

Install on every node of cluster

Setup ‘temp directory’ to a path that is same on all nodes. For example, c:\temp

Linux:

Install on a single node, on a shared drive.

Setup ‘temp directory’ to a path that is same on all nodes. For example, /tmp

Ensure that the product is available using the same path on all nodes

Permissions:

All users of the cluster should have read/write permissions to temp directory

All users should have read/execute permissions to installation directory

When a desktop scheduler GUI is run the same node as the job submission node, no other configuration is necessary: installation is sufficient. You select the scheduler through the desktop GUI. You need to ensure that scheduler commands are available in the path before you launch desktop.

Note	There is no need to install RSM unless the you are using the scheduler GUI on a post processing node that is different from the job submission node. In this case, RSM must be configured with the scheduler type and path. A post processing nodes is a node in the cluster that can run the Ansys Electromagnetics desktop in graphical mode. A job submission node is a node in the cluster in which job submission commands are available.

Turn OFF firewall between cluster nodes.

Scenario 1: The post-processing node and job-submission node roles are served by distinct machines.

In this case, perform the following configuration:

The job-submission node should be configured to run the RSM service, which serves as a proxy to scheduler. The RSM Service should be running as 'root' in order to facilitate jobs running using the credentials of the job's owner. A **configuration file in the RSM installation folder should be edited** to specify information regarding the scheduler that manages jobs on this cluster. A block labeled 'Scheduler' must be included within the 'AnsoftCOMDaemon' block. This block contains two string entries:

- SchedulerName: this contains the unique part of the scheduler proxy library name
- ConfigString: this contains a scheduler specific configuration string

The case of the SchedulerName string is significant on Linux because Linux file names are case sensitive. The case of the SchedulerName string is not significant on Microsoft Windows. In Ansys Electromagnetics Suite, the possible scheduler names are: lsf and sge. The ConfigString entry is a scheduler specific configuration string, described below.

In addition, the AnsoftRSMService must be started with appropriate environment variables set. Generally, the environment variables must be set the same as they would be set for using the scheduler via command lines.

SGE Details

For SGE, the ConfigString entry must contain the search path for the SGE commands. It may contain a single directory, the directory containing the SGE commands. Alternatively, it may be a

path, with directories separated by the colon character ":", where the SGE command directory appears before any other directory containing files with the same name as any SGE commands.

Example ansoftrsmservice.cfg configuration file:

```
$begin 'AnsoftCOMDaemon'
$begin 'Managed COM Servers'
$end 'Managed COM Servers'
$begin 'Scheduler'
'SchedulerName'='sge'
'ConfigString'='/opt/sge6.2u4/bin/lx24-amd64'
$end 'Scheduler'
$end 'AnsoftCOMDaemon'
```

Scenario 2: The post-processing node and job-submission node roles are served by the same machine.

The **Select Scheduler...** command (as described in the Job Management User Interface for SGE section) is used to gather details about the scheduler. In this case, the Desktop process should be started in an environment suitable for submitting jobs to the scheduler.

The environment should be configured so that all SGE commands are found using the standard search path. In particular, search for the following commands in the search path should result in the SGE command being found: "qsub", "qdel", "qstat", and "qconf". No other command with the same name should appear before the SGE command in the search path.

Related Topic

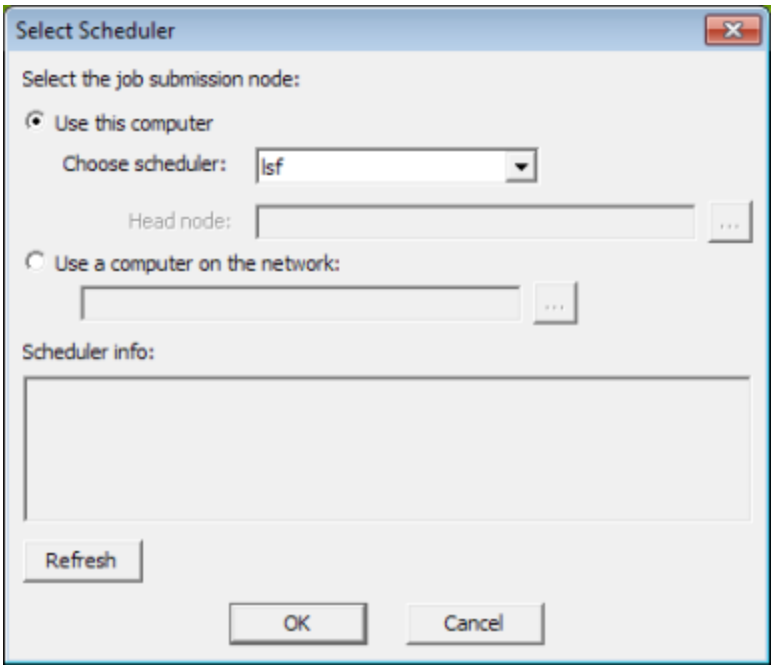
Job Management User Interface for GE

Command Line Enhancements for Ansys Electromagnetics Desktop Products

Job Management User Interface for SGE

The Job Management UI is accessed by running Ansys Electromagnetics product Desktop on the designated 'Postprocessing node' of the cluster. The Desktop provides UI commands for Scheduler selection, Job submission and Job monitoring/control. You access the Scheduler User Interface by clicking **Tools>Job Management>Select Scheduler.**

Before you can use **Submit Job**, you must click **Select Scheduler** as the one-time initial step. This opens the **Select Scheduler** dialog.



Specify the following parameters:

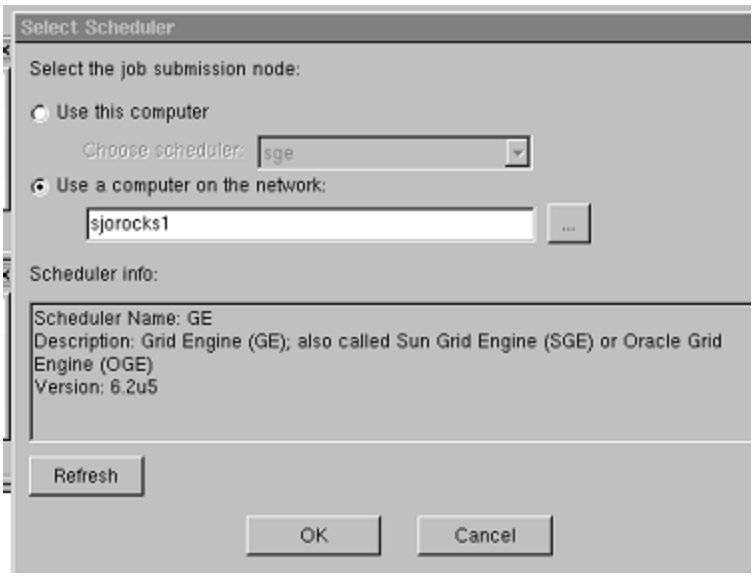
- Job submission node: this is the node on the cluster where scheduler commands (such as SGE's qsub) are allowed to run.

  Choose **Use this computer** if scheduler commands are enabled on the post-processing node.

  Choose **Use a computer on network** if the cluster is configured in a manner as to disallow job-submission from the post-processing node. Specify node name appropriately.

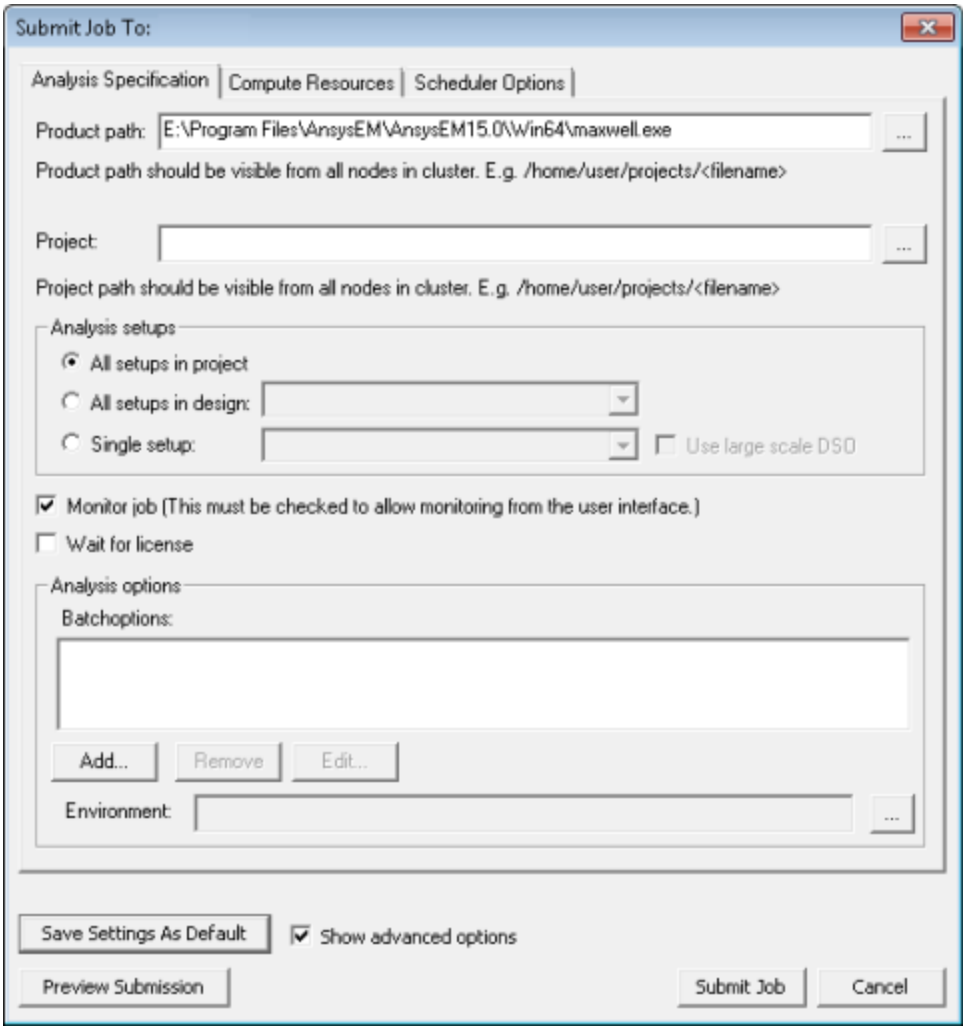
  Pre-requisites: For this choice to work, the job-submission node must already be configured with a running RSM service, as documented in Installation of Ansys Electromagnetics Tools on GE.
- Scheduler: Available choices, depending on your installations, are: RSM, lsf, Windows HPC, and SGE. It is also possible for you to integrate a custom scheduler into this UI, through a scheduler proxy. When this is done, more choices will be available in the combo-box, one per custom scheduler proxy that is deployed in the installation.

After specifying the job submission node, you can click **Refresh**. The scheduler information is then listed in the Scheduler info text field.



Once you select a scheduler, you can access the interface for job submission, monitoring and control. Click OK to close the dialog.

You access the Job submission UI by clicking **Tools>Job Management>Other Schedulers>Submit Job...**. This command launches a multi-tab Submit Job To: dialog similar to the one shown below.



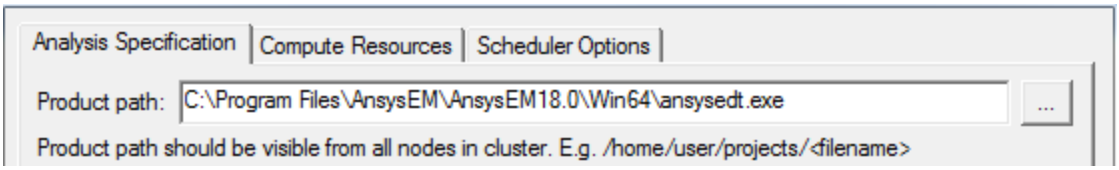
- **Analysis Specification--**specify the Product path, Project name, the setups, and analysis options such as batchoptions, or, for advanced users, Environment variables. If you select the Analysis or Optimetrics setup, the Analysis Specification is pre-populated.
- The **Compute Resources** tab specifies whether to use automatic settings (if available for the design type), and how to select specific resources from the available pool (for example, ParallelEnvironment is an SGE parameter).
- The **Scheduler Options** tab has analysis-independent parameters specific to the job, such as name, priority.

Analysis Specification tab for GE

This tab lets you specify the following:

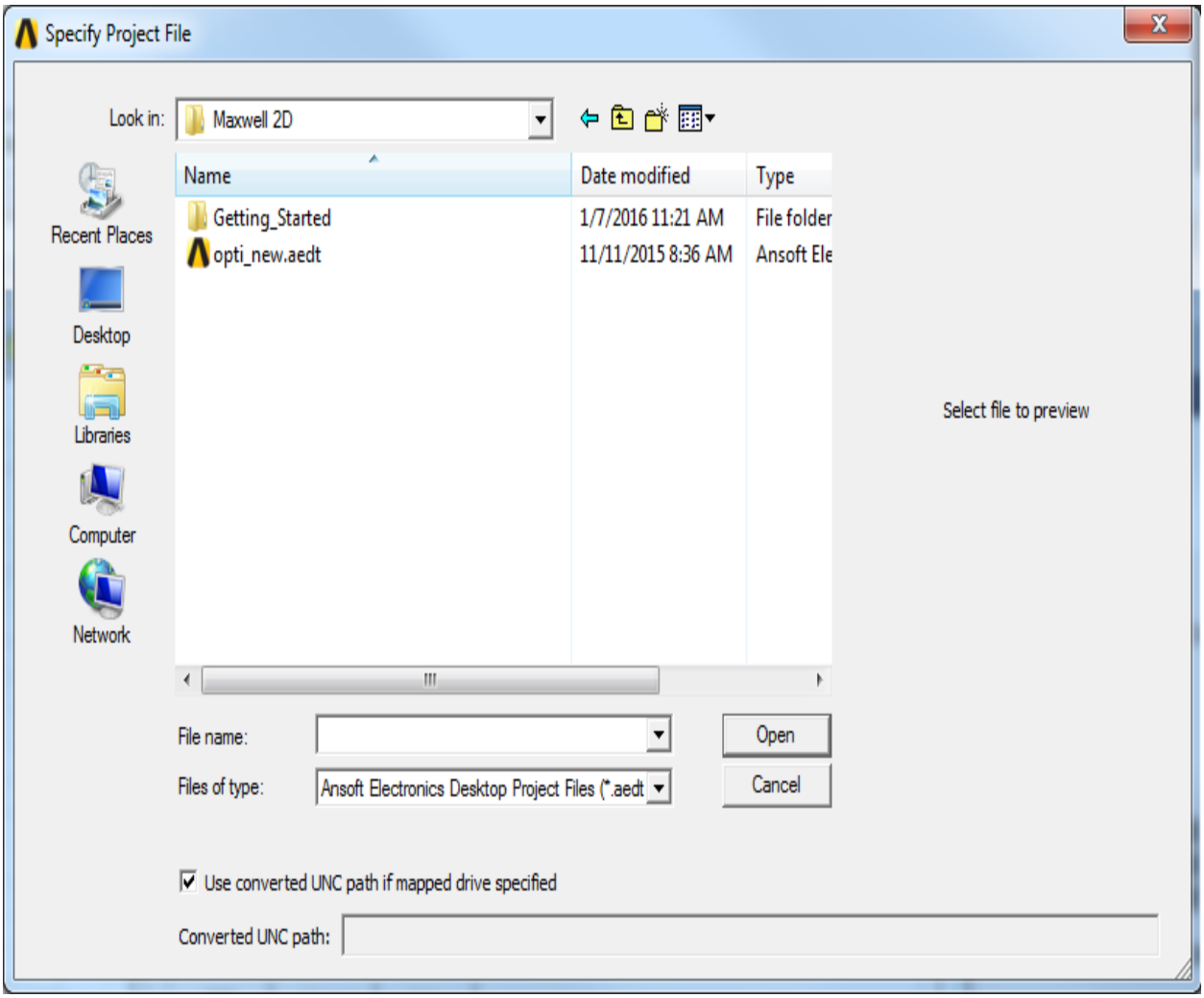
Product Path:

You can specify the product path.



Project:

You can use the ellipsis button [...] to use a navigation window to browse. The path should be visible to all nodes in the cluster. The Project can be an archive.

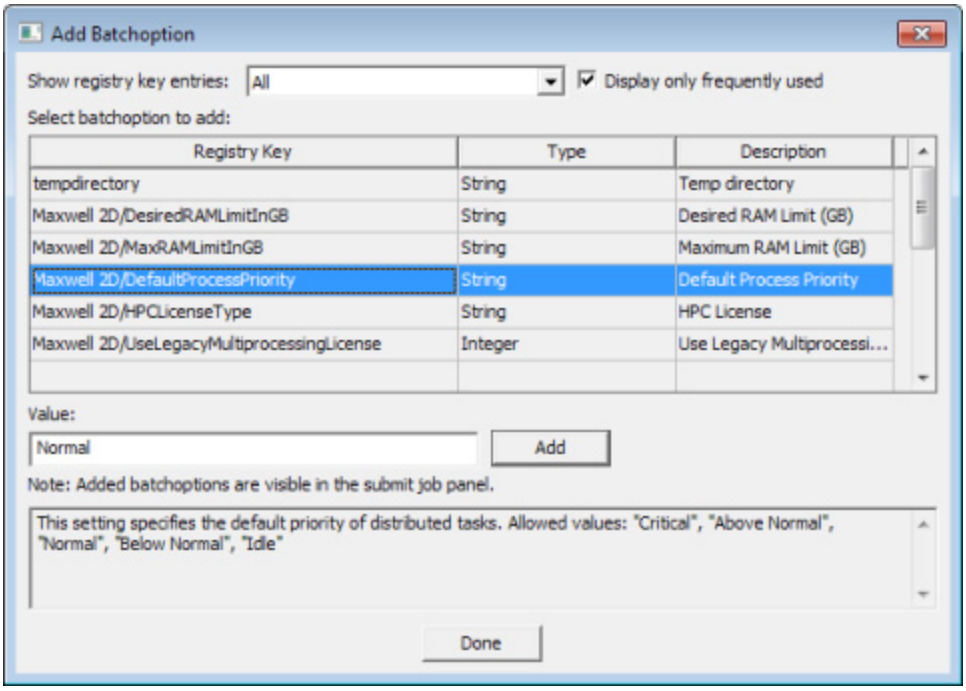


Analysis Setups:

Specify All setups in the project, all in a design, or a single setup. If the setup includes a parametric sweep, the **Use large scale DSO** check box is enabled.

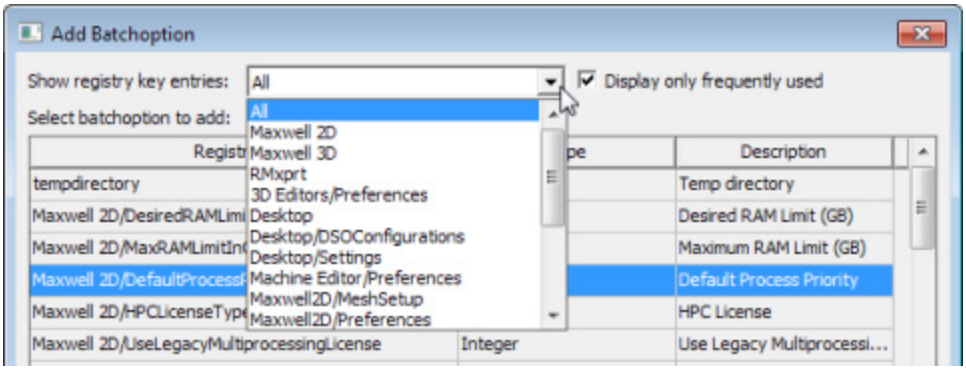
Analysis Options:

- You have check boxes for whether to Monitor Job through the GUI, and whether to Wait for license.
- You can also specify Batchoptions. Click the **Add...** button to open a dialog for selecting the Batchoptions.



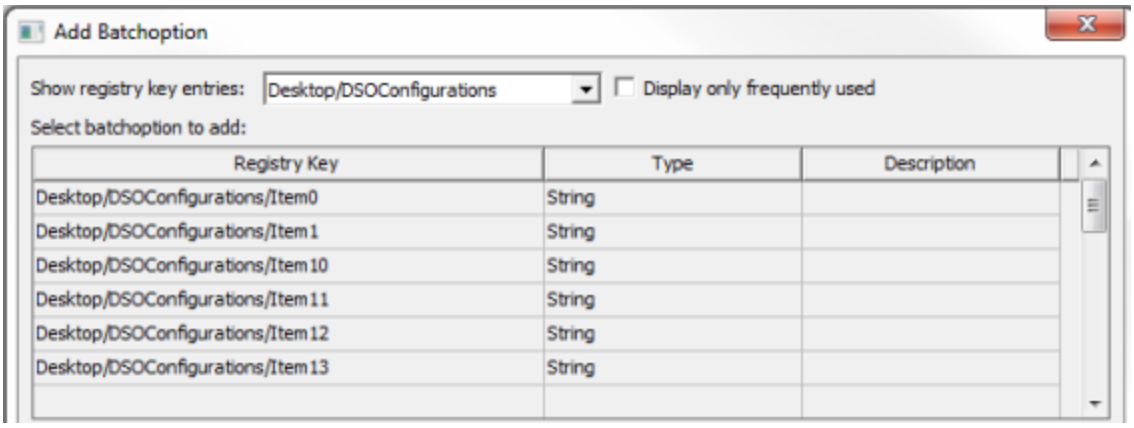
The lower Value field shows the legal values for the selected registry key. You can type the desired value into the upper text field under value. Click the **Add** button to accept the selection of the registry key with the specified value. Click **Done** close the **Add Batchoption** dialog.

At the upper left, a drop down menu lets you specify which registry key categories to display, whether All, or selected category.

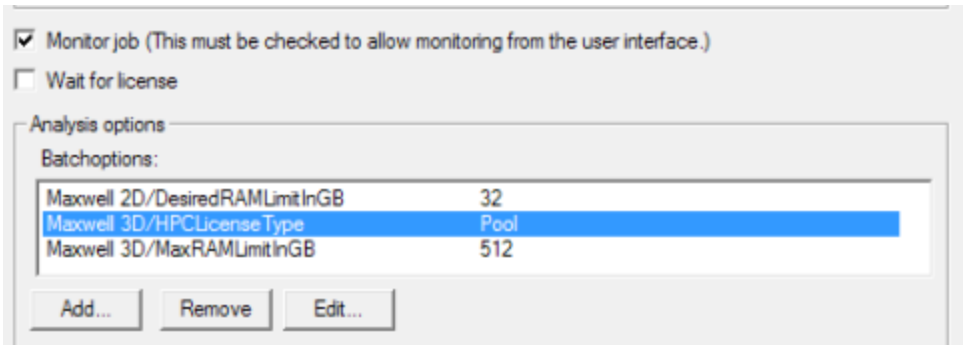


A check box lets you choose between displaying only frequently used entries (the default), or by unchecking, all options available for the selected group.

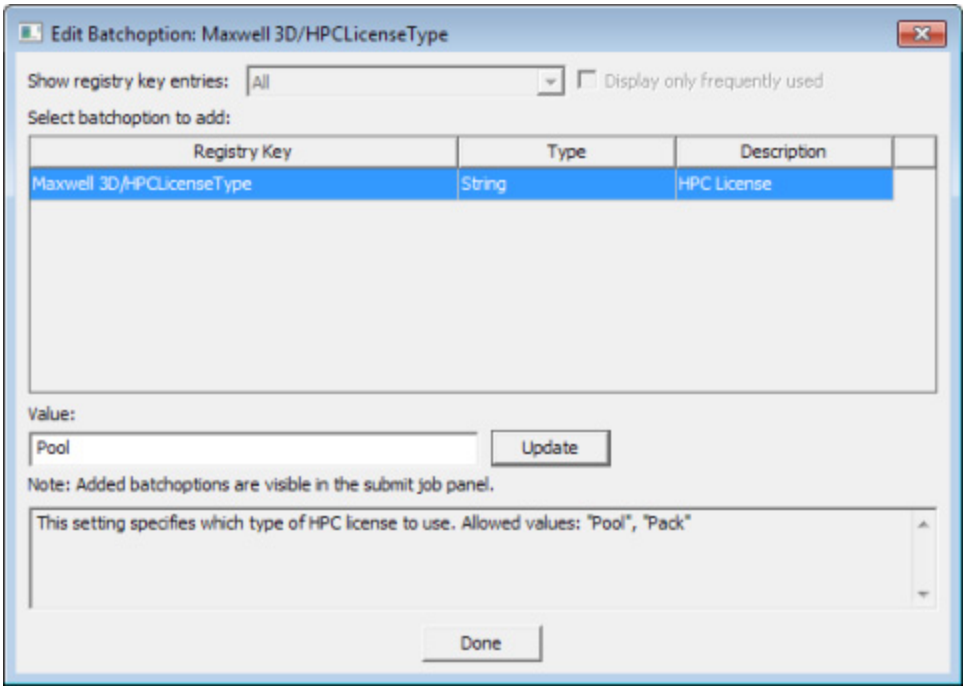
Note that for Large Scale DSO problems, you also have the option to specify predefined configurations.



The added registry keys and values are listed in the Batchoptions field of the **Submit Job** dialog.



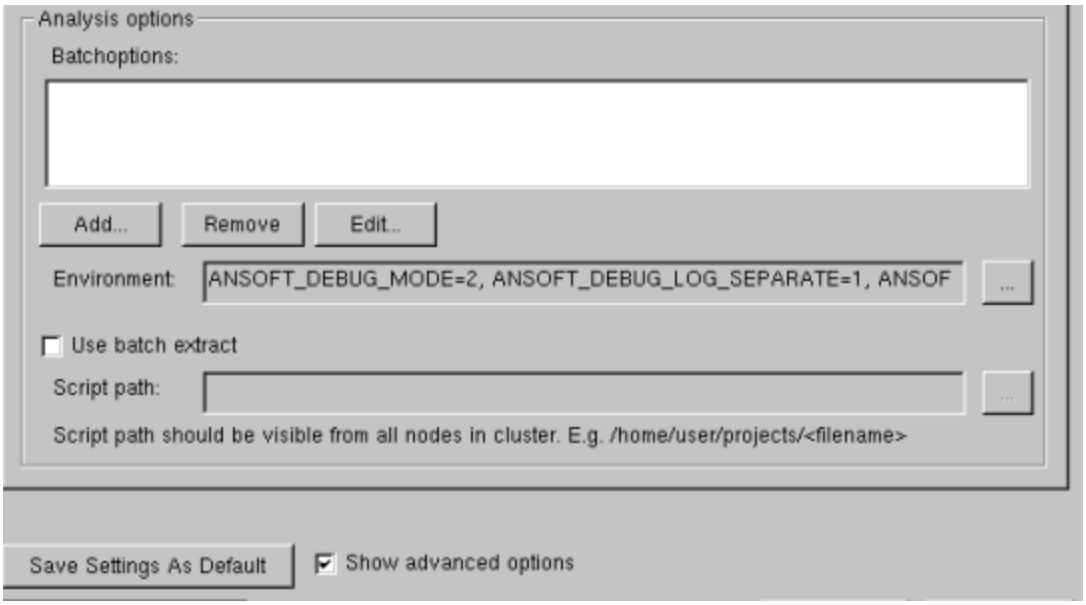
Selecting from the list enables buttons for removing or editing registry key values. Selecting a registry key and clicking **Edit...** opens the **Edit Batchoptions** dialog.



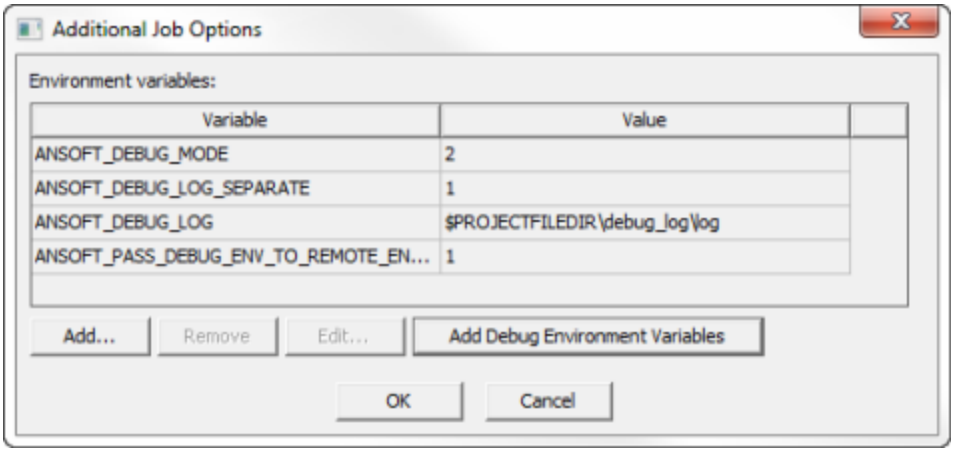
Here you can edit the Value field and update the value for the selected registry key.

Environment:

If you check the **Show advanced options** check box, you can see the Environment field.



This permits specifying Environment Variables. Click the ellipsis button [...] to open the **Additional Job Options** dialog box.

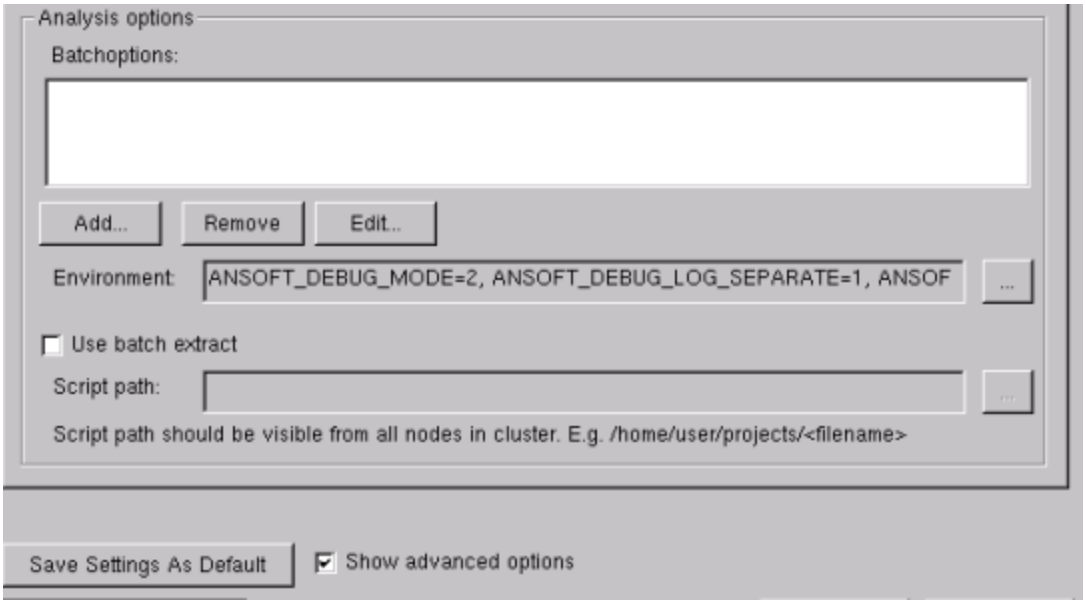


The **Add** button permits you to specify your environment variables. Once, added, you can select **Remove** or **Edit**. The **Add Debug Environment Variables** button adds variables of use in working with support.

- The **Save Settings as Default** button lets you save a current set of values as defaults the next time you invoke the Scheduler GUI. This can simply subsequent job submissions.

Use Batch Extract for Grid Engine

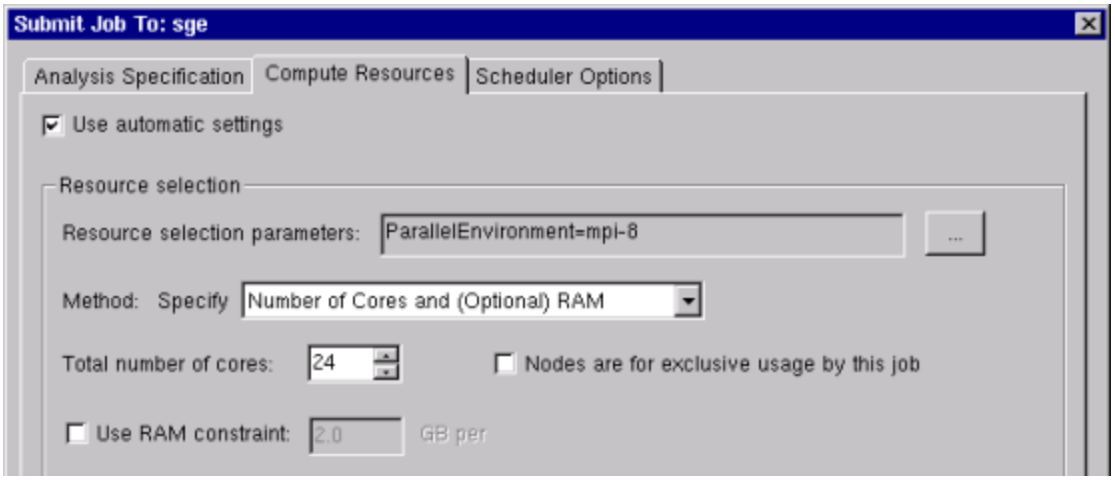
Selecting **Show advanced options** for Grid Engine also show the **Use batch extract** fields.



See the discussion on Running Maxwell from a Command line for a discussion of the solve information available through batch extract.

Compute Resources Specification Tab

This tab lets you provide Resource selection and Job parallelization parameters. Once you have specified parameters, you can Save Settings as Defaults, Preview a Submission, and Submit a job.

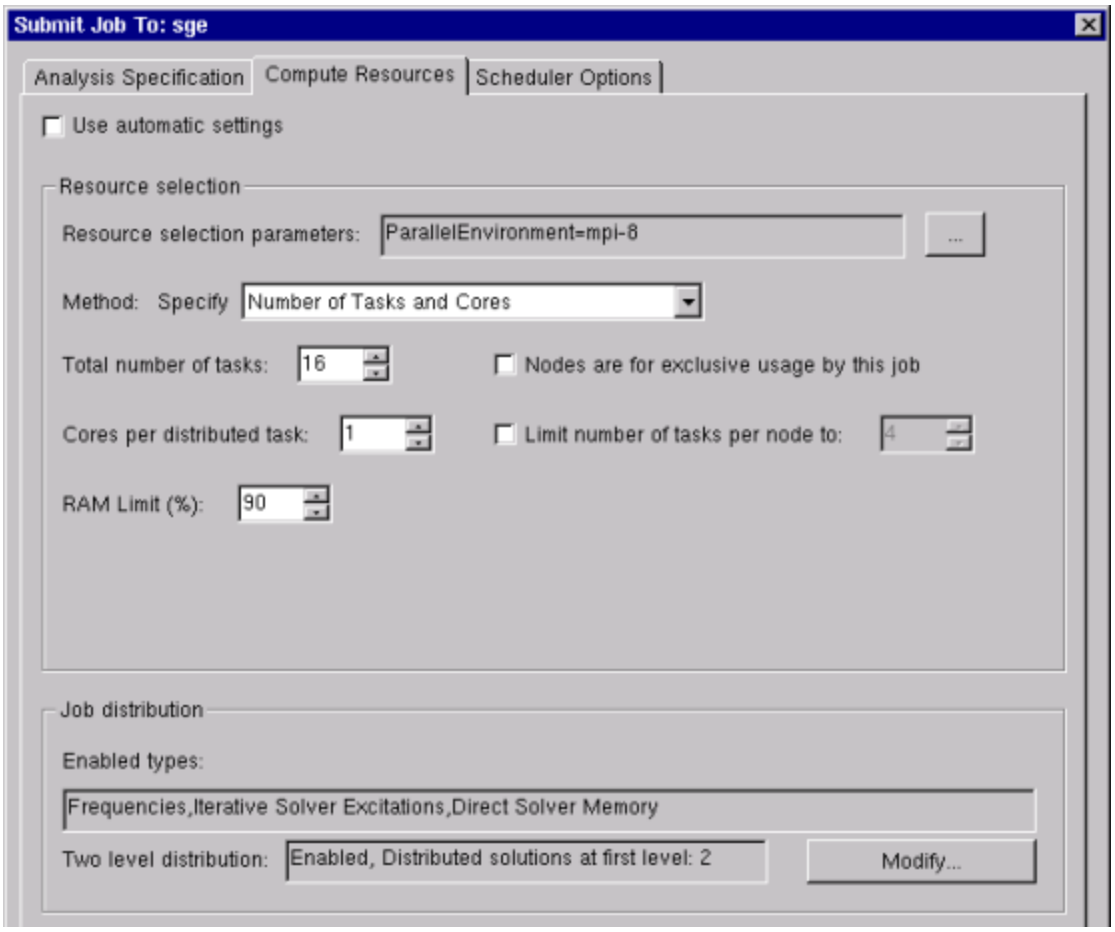


You can also populate this page yourself by specifying Resource selection, Job Parallelization parameters, and Job distribution parameters.

This tab lets you provide Resource selection and Job parallelization parameters. You can elect to **User automatic settings** for job distribution, specify a Method and resource selection parameters, or job distribution parameters.

Note	If you select Use automatic settings, the solver does not support automatic distribution of variations. The Use automatic settings option does not support Optimetrics variations. It does distribute frequencies, domains, and use of multiple level domains.

You can also uncheck **Use automatic settings**. Doing so requires you to also specify Cores per distributed task and enables the Job distribution fields.

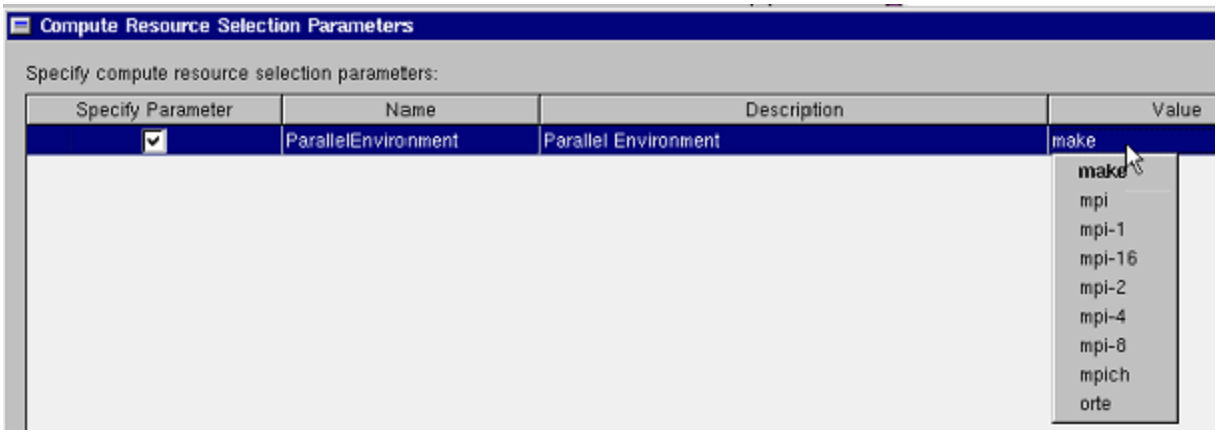


Resource Selection:

If you do not specify parameters for resource selection, SGE may submit jobs to any machine in the entire pool that is available.

- Resource selection parameters.

  Clicking the ellipsis button [...] opens a dialog for parameters specific to SGE, in this case **ParallelEnvironment**. The Specify Parameter check box in the dialog enables a parameter, and you can select the Value for the ParallelEnvironment parameter from a drop down menu.



If you don't specify a parameter, the scheduler handles the situation.

- Specify node list

  Checking Specify node list enables the field for specifying a node list. In a computing environment where the available cores are not uniform, you can use this to have control over which resources your job will use. If your Analysis configuration contains a node list, you can use Populate this Page from Analysis Configuration.

Memory resource behavior is dependent on the Grid Engine version as well as the particular scheduler settings. The output of the "qconf -sc" command shows all of the complexes available to the Grid Engine schedulers. For a complex to be considered valid for memory resource selection, it must be of type "MEMORY", have relop "<=", be request-able (could be "forced" as well, instead of "YES"), and be consumable. At least one valid memory complex must be available to submit jobs using the Automatic Cores and RAM method.

Univa 8.3 introduced a new column to the complexes listing. For Univa 8.3 and later, there is an additional column for whether resources are available to a preempting job after preemption of a running job. It is up to the cluster administrator to determine the appropriate Available After Pre-Emption (aapre) setting for memory complexes. This setting is ignored for memory complex validation.

Determining the correct memory complex by default is error-prone. Because the correct choice of memory complex can vary from cluster to cluster, the memory complex selection is now exposed by default under the compute resource selection parameters, allowing the user to make the selection without having to set an environment variable. Only complexes that have been validated (meeting the requirements specified above) can be selected. The cluster administrator (or someone who has knowledge of the specifics of the cluster in use) should be able to determine the correct memory complex to use for Ansys Electronics jobs. To disable exposure of this selection option, the following environment variable can be set to "0":

ANSOFT_SGE_ENABLE_MEM_RES_ATTRIB

Method

For the Method, use the drop down to select Number of Cores and (Optional RAM) or Number of Nodes and Cores.

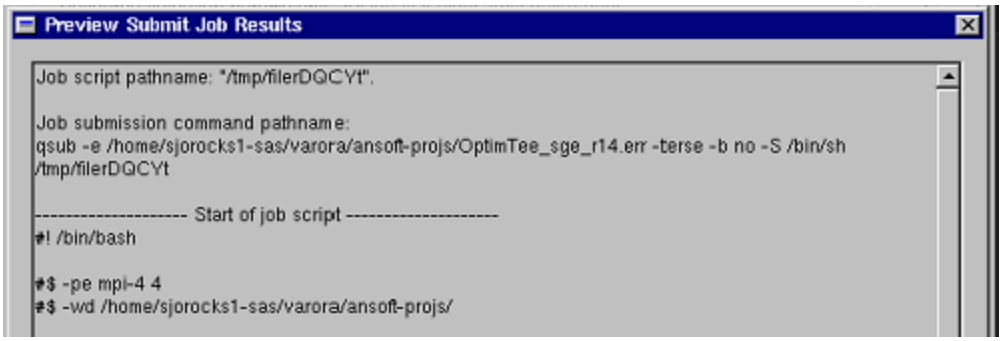
Total Number of Cores

You can specify a Total number of cores, and whether the Nodes are exclusive to the job.

RAM Constraint and RAM Limit

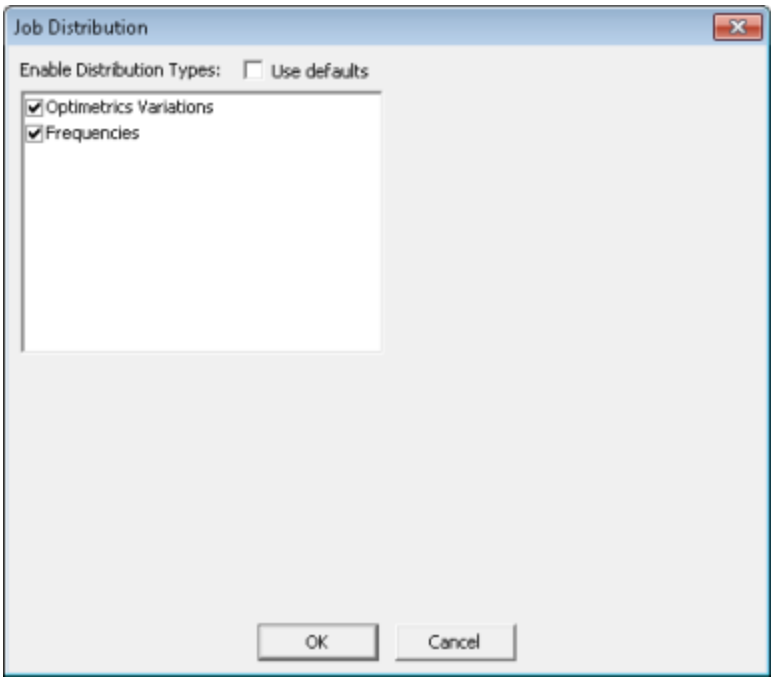
If you check Use RAM Constraint, you can specify a GB RAM constraint. You can also specify a RAM Limit as a percent.

In response to a set of minimal constraints, the Scheduler may increase the resources assigned beyond the minimal values in order to meet the full set of requirements. Notice that a preview of the Submit Job Results shows the number of resources assigned, and that the scheduler generated code includes an MPI specification.



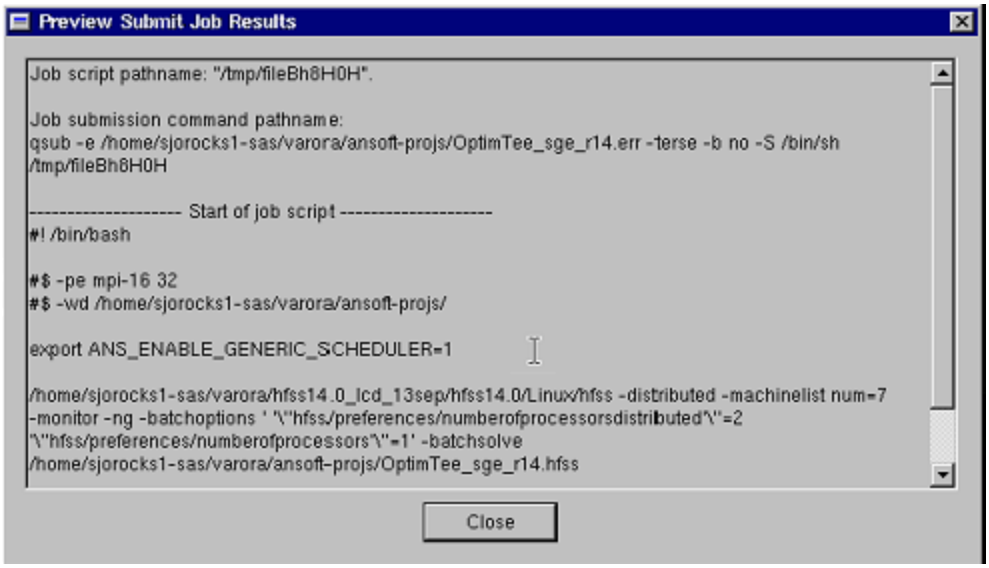
Job Distribution

- Enabled types, such as Optimetrics Variations, Frequencies, etc.

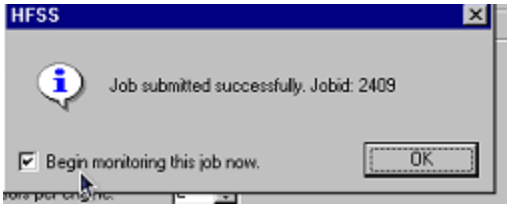


Preview submission

This opens a screen showing the qsub command to be used to submit the job.



Submit job actually sends the batch command to the SGE cluster.A dialog reports a successful submit and presents a check box for monitoring.

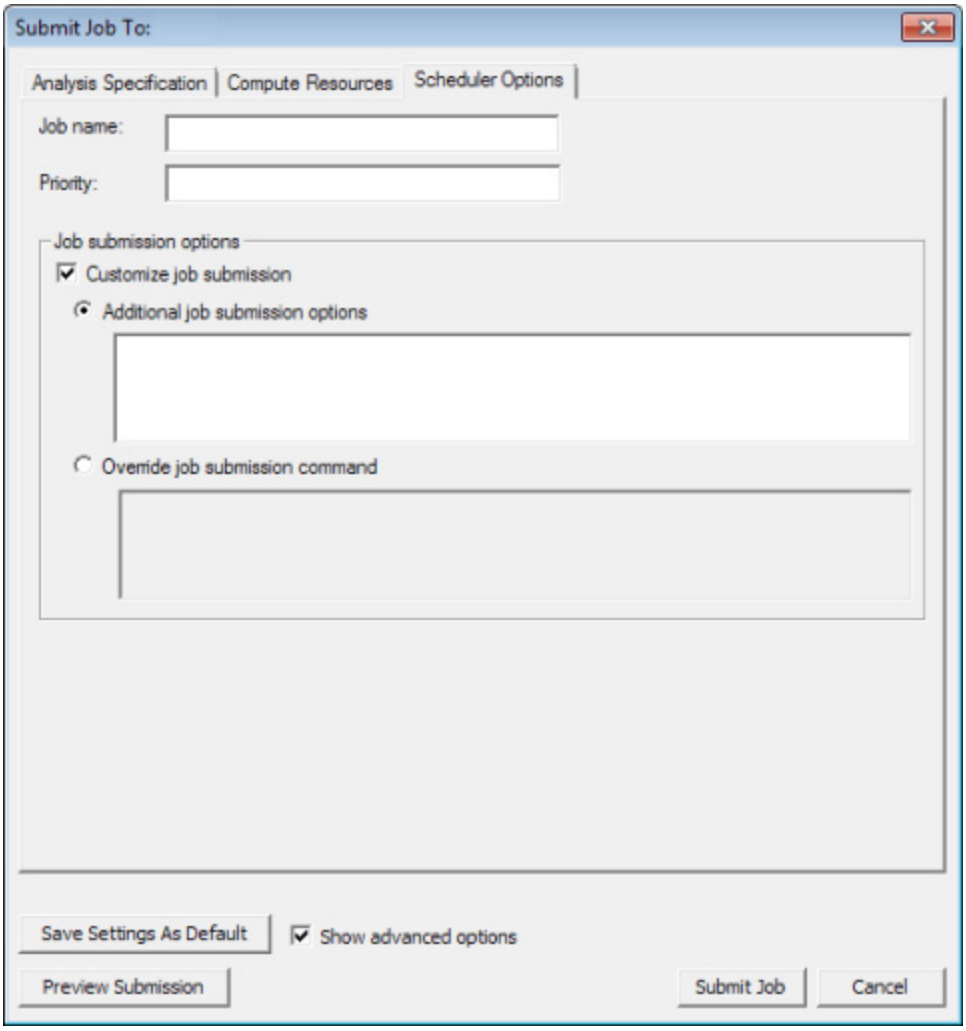


Scheduler Options Tab

This tab lets you give Job name and Priority

If you check **Show advanced options**, you can also specify **Job submission options**.

The Job submission options permit you to customize a job submission in terms of addition job submission options, or override a job submission command.



When the "Override job submission" radio button is checked, the user specified options replace most of the job submission options, whereas when the "Additional job submission options" radio button is checked, the user specified options are appended to the bsub command.

Text in the enabled field is appended to the bsub command. You can see the effects of any custom additions by clicking **Preview Submission**.

Related Topics

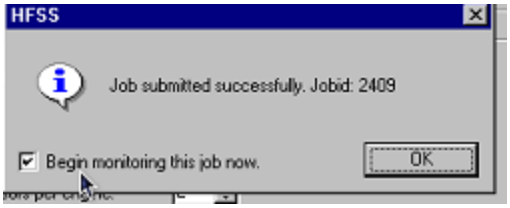
Integrated Job Monitoring for Job Management Interface for SGE

Scheduler Proxy Interfaces

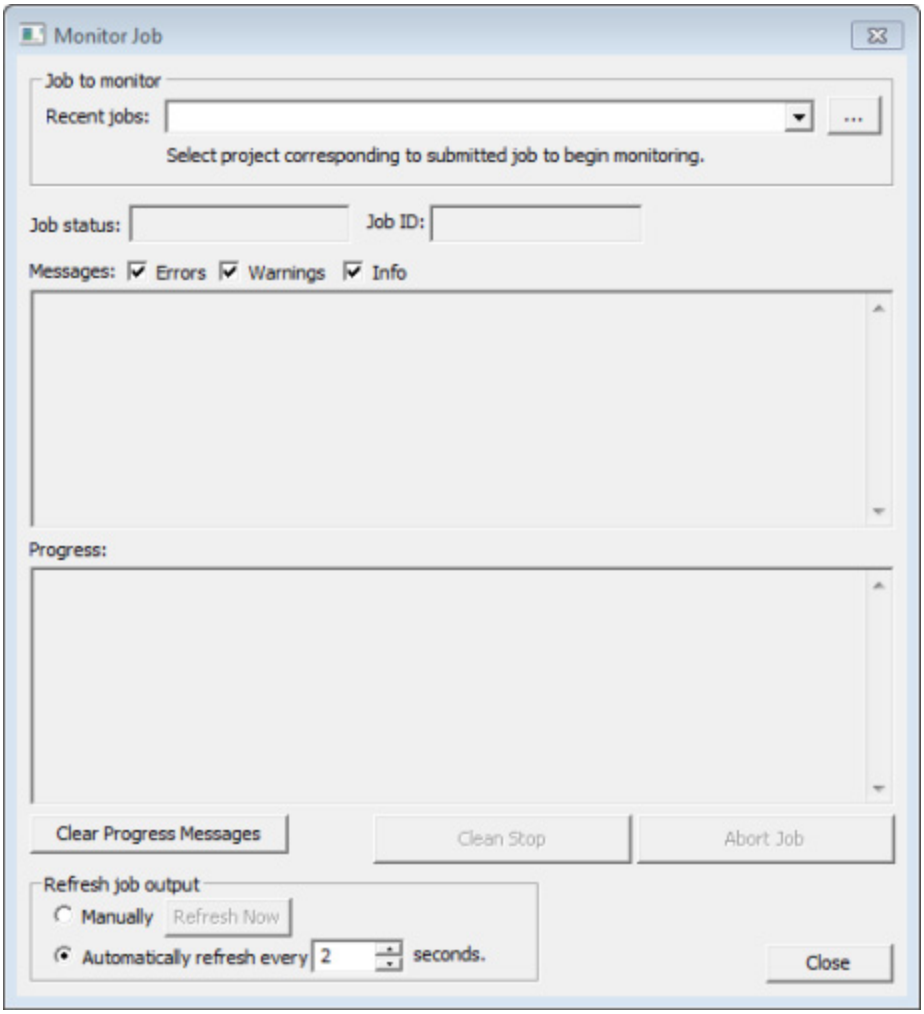
Integrated Job Monitoring for Job Management Interface for SGE

The job monitoring/control dialog is launched through the command **Tools>Job Management>Monitor Jobs... o**r by checking **Begin monitoring this job now** in the

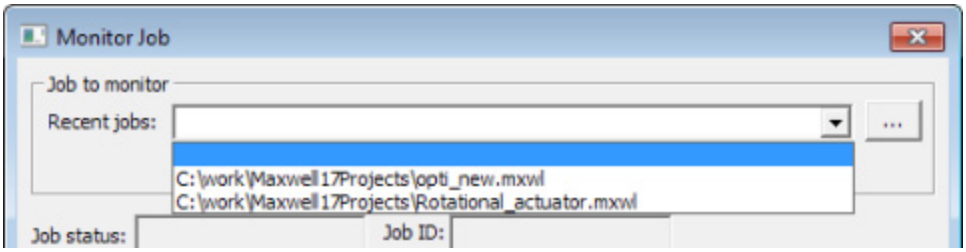
information window reporting successful job submission.



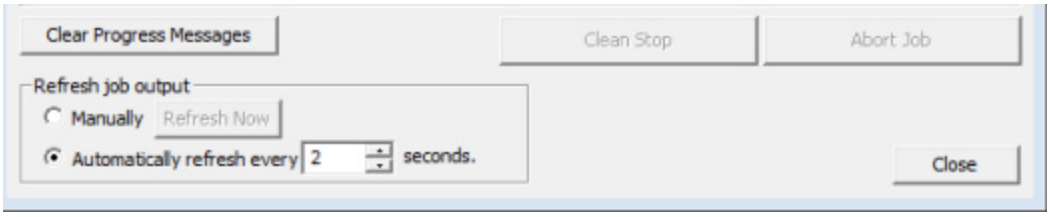
In response to either invocation, the **Monitor Job** dialog opens:



In this dialog box, you can select the job to monitor from a pull down list of recent jobs.



The lower left has corner of the dialog has options for manual refresh or to Automatically refresh every specified number seconds. After refresh, the messages for the job are displayed.The lower left has corner of the dialog has options for manual refresh or to Automatically refresh every specified number seconds. After refresh, the messages for the job are displayed.



You can choose to **Clear Progress Messages**. The **Clean Stop** button lets you stop the simulation cleanly between time steps.

The **Abort Job** button lets you abort a job.

Related Topics

Scheduler Proxy Interfaces

Job Management User Interface for SGE

SGE Command Line Interface

SGE Commands for Information About Jobs and Cluster Configuration

The following SGE commands are especially useful for getting information about the cluster configuration or for getting information about running or completed jobs. This list only contains a few of the most common commands. Consult the SGE man pages for a complete list and more details.

`qconf -help`: The first line displays the SGE version

`qacct -j` *job-id*: Displays a log of the completed job with id job-id (if accounting is enabled)

`qstat -j` *job-id*: Displays a log of the running job with id job-id

`qconf -sc`: Show all complex attributes

`qconf -spl`: Show a list of all parallel environments

`qconf -sp` *pe-name*: Show details of parallel environment named pe-name

`qconf -sql`: Show a list of all queues

`qconf -sq` *queue-name*: Show details of queue named queue-name

`qconf -sconf`: Show configurations

Submitting Ansys Electromagnetics SGE Batch Jobs

The SGE qsub command may be used to submit Ansys Electromagnetics jobs. Typical command formats are:

```
qsub qsub_argsansysEM_exeansys_args
qsub qsub_argsjob_script
qsub qsub_args[ -]
```

where:

- *qsub_args* are the options of the qsub command,
- *ansysEM_exe* is the pathname of the Ansys Electromagnetics tool executable to launch,
- *ansys_args* are the arguments to the Ansys Electromagnetics tool command, and
- *job_script* is a shell script containing the Ansys Electromagnetics desktop command to run.

In the first format, the Ansys Electromagnetics desktop command and its arguments are specified on the **qsub** command line. In the second format, the pathname of a shell script containing the Ansys Electromagnetics desktop command and its arguments is specified on the **qsub** command line. In the third format, the command is omitted or replaced with a hyphen; this indicates that the command or script will be taken from stdin.

Quoting Ansys Electromagnetics Command or Arguments for SGE

If the Ansys Electromagnetics tool executable pathname (*ansys_exe*) or any of the arguments of the Ansys Electromagnetics tool command (*ansys_args*) contain characters which are interpreted by the command shell, then these special characters must be properly quoted to ensure that the correct command is launched by SGE. This is especially important when using the first form of the **qsub** command, as the Ansys Electromagnetics desktop command is processed by the shell twice in this case. It is processed by the shell when the **qsub** command is processed, and again when the job is started.

Serial SGE Batch Jobs

In general, Ansys Electromagnetics batch jobs may be submitted as SGE serial jobs without any special considerations.

See Monitoring Ansys Electromagnetics SGE Batch Jobs for options for monitoring Ansys Electromagnetics batch jobs.

Parallel SGE Batch Jobs

When an Ansys Electromagnetics batch job is run as an SGE parallel job, the SGE scheduler will select the hosts for the distributed analysis job, and start the desktop process on one of these hosts. The desktop process will obtain the list of hosts from the SGE scheduler, and start analysis

processes, as needed, using the SGE scheduler facilities. To run an SGE parallel job, the job must be submitted to an SGE parallel environment (PE).

If the qmaster tcp port is not configured as a service, but rather via the environment variable SGE_QMASTER_PORT, this variable must be set in the Ansys Electromagnetics batch job environment. This is needed because the Ansys Electromagnetics desktop uses the "qrsh -inherit" command to launch engine processes.

See Monitoring Ansys Electromagnetics SGE Batch Jobs for options for monitoring Ansys Electromagnetics batch jobs.

Setting Up an SGE Parallel Environment (PE)

To allow Ansys Electromagnetics batch jobs to distribute analysis engines to multiple hosts, the job must be run in a parallel environment (PE) in which the control_slaves parameter is set to TRUE. This setting is required to allow the Ansys Electromagnetics desktop to start analysis engines on hosts other than the local host, i.e., the host where the Ansys Electromagnetics desktop is running.

Here is a sample parallel environment configuration:

```
pe_name ans_test1
slots 999
user_lists NONE
xuser_lists NONE
start_proc_args /bin/true
stop_proc_args /bin/true
allocation_rule $round_robin
control_slaves TRUE
job_is_first_task FALSE
urgency_slots min
accounting_summary TRUE
```

The user_lists and xuser_lists parameters are ACLs (access control lists) used to control which users have permission to use the parallel environment. The user_lists setting gives permission to use the PE. The xuser_lists setting denies permission to use the parallel environment. The xuser_lists settings override the user_lists settings.

The start_proc_args and stop_proc_args parameters contain the pathname and arguments for the parallel environment startup and shutdown scripts. No startup or shutdown scripts are needed for parallel Ansys Electromagnetics batch jobs. The setting /bin/true may be used as the value for these scripts; this utility does nothing and returns an exit code indicating success (0).

The parallel environment allocation_rule parameter will affect how the analysis engine tasks are distributed across the hosts allocated to the job. The $round_robin setting distributes the tasks across the hosts in a round robin fashion, resulting in the load being relatively evenly distributed over all of the hosts. The $fill_up setting allocates all slots on a host before distributing the tasks to

another host; the result is that most hosts are either fully utilized or completely unused. See the sge_pe man page for other settings for this parameter.

The control_slaves parameter must be set to TRUE, as described above.

The job_is_first_task parameter also affects how tasks are allocated. When submitting a job to run in a parallel environment, the number of parallel tasks, n, is specified on the command line. If this setting is TRUE, then the job process is considered one of the tasks, and only (n-1) additional tasks are allocated to the job. If the setting is FALSE, then the job process is not considered to be one of the tasks, and n additional tasks are allocated for the job.

See the sge_pe man page for more information about these and other PE parameters.

A parallel environment does not run tasks directly. Instead, the tasks are distributed to queues associated with the parallel environment. In order to complete the setup of a parallel environment, one or more queues need to be associated with the parallel environment. The queue pe_list parameter is used to specify the parallel environments (PEs) supported by the queue. This is an important step; **if no queues support a given PE, then jobs submitted to that PE will not run**.

Parallel Batch Job Command Line Considerations

The number of engines run on a host will depend on the total number of distributed engines, and the number of hosts allocated to the job. The memory required on a host depends on the number of engines running on the host and on the memory needed for each engine. The **qsub** command **-l resource=***value*,... or **-q** *queue_list* command line options specify that the parallel batch job run on machines with sufficient memory and other resources.

Related Topics

Monitoring Ansys Electromagnetics SGE Batch Jobs

Ansys Electromagnetics Desktop -monitor Command Line Option for SGE

Example SGE qsub Command Lines

Issue with MainWin Core Services for SGE

What a Scheduler Does

Recommended Practices for SGE Clusters

Monitoring Ansys Electromagnetics SGE Batch Jobs

The suggestions below may be used for SGE serial jobs and for SGE parallel jobs.

SGE qstat Command

The SGE **qstat** command may be used to display information on jobs and queues. If the **-j [***job_list***]** option is included, then information on jobs is displayed. If the **-j [***job_list***]** option includes a job list, then the displayed information is limited to the jobs in the job list.

The **-u***user*,... option limits the output to jobs associated with users in the user list. If the **-u***user*,... option is not specified, then information on queues or jobs of the current user are displayed.

The **-t** option displays extended information about the subtasks of each displayed job. This is equivalent to the **-g t** option. The **-r** option displays extended information about the resource requirements of the displayed jobs.

See the SGE manual pages for more information.

Related Topics

Integration with Sun Grid Engine (SGE)

Job Management User Interface for SGE

Integrated Job Monitoring for Job Management Interface for SGE

Ansys Electromagnetics desktop -monitor Command Line Option for SGE

Example SGE qsub Command Lines

Issue with qrsh (SGE)

Issue with MainWin Core Services for SGE

Ansys Electromagnetics Desktop -monitor Command Line Option for SGE

The **-monitor** command line option enables batch job output to the standard output and standard error streams. The warning, info, and progress messages are sent to the standard output stream. The error and fatal messages are sent to the standard error stream.

The SGE scheduler redirects the standard output and standard error streams of batch jobs to files specified in the **qsub -o [[***hostname***]:]***path*,... and the **-e [[***hostname***]:]***path*,... command line options, respectively. If either option is not specified, then the associated stream is redirected to the default file pathname.

The **qsub -j y[es] | n[o]** controls whether the standard output and standard error streams are merged. If the y or yes value is specified, then the standard error stream is merged into the standard output stream. If the **-e** *host_and_path* option is also specified in this case, the *host_and_path* setting is ignored. If the n or no value is specified, or if this option is not specified, then the standard error stream and standard output stream are not merged.

You can monitor the progress of a job by checking the standard output file for progress, info and warning messages, and checking the standard error file for error and fatal messages.

Related Topics

Integration with Sun Grid Engine (SGE)

Monitoring Ansys Electromagnetics SGE Batch Jobs

Example SGE qsub Command Lines

Recommended Practices for SGE Clusters

Issue with qrsh (SGE)

Issue with MainWin Core Services for SGE

Example SGE qsub Command Lines

All of the following examples show how to submit Linux Maxwell jobs on SGE, but similar command lines will work for other Ansys Electromagnetics products.

Serial job using command line:

```
qsub -b y /AnsysEM/v231/ansysedt -ng -BatchSolve
~/projects/OptimTee.aedt
```

- The **-b y** option indicates that electronics desktop is launched directly from the command line, instead of using a script.
- No queue is specified, so the default queue will be used

Serial job with a hard runtime limit of 15 minutes:

```
qsub -b y -l h_rt=00:15:00 /opt/AnsysEM/v231/ansysedt
-ng -BatchSolve ~/projects/OptimTee.aedt
```

- The -l h_rt=00:15:00 option indicates that this job has a "hard" runtime limit of 15 minutes.

Serial job using a script, with a runtime limit specified in the script:

```
qsub ~/sge/scripts/OptimTee.csh
```

- The **-b y** option is absent, so the script ~/sge/scripts/OptimTee.csh will be run when the job starts.
- The script file OptimTee.csh may contain SGE directives in addition to the command(s) to run. In this example, a directive with a hard runtime limit if 15 minutes is included in the script.

Script file contents:

```
#!/bin/csh
#$ -l h_rt=00:15:00
/opt/AnsysEM/v231/ansysedt -ng -BatchSolve
~/projects/OptimTee.aedt
```

- The SGE directive #$ -l h_rt=00:15:00 is equivalent to including -l h_rt=00:15:00 on the qsub command line.

Distributed processing job using 4 engines:

```
qsub -b y -pe pe1 4 /opt/AnsysEM/v231/ansysedt
-ng -BatchSolve -Distributed -machinelist num=4
~/projects/OptimTee.aedt
```

- The **-b y** option indicates that maxwell is launched directly from the command line, instead of using a script.

- The **-pe pe1 4** command_line option indicates that this is a parallel job running under the pe1 parallel environment, and that 4 cores or processors are allocated to this parallel job.
- The "-machinelist num=n" option is now required for batch jobs.
- The **-Distributed** option indicates that this is a DSO job, so that multiple engines will be started. Because 4 cores are allocated to the job, the job will run 4 engines.

Related Topics

Integration with Sun Grid Engine (SGE)

Monitoring Ansys Electromagnetics SGE Batch Jobs

Ansys Electromagnetics Desktop -monitor Command Line Option for SGE

Recommended Practices for SGE Clusters

Issue with qrsh (SGE)

Issue with MainWin Core Services for SGE

Recommended Practices for SGE Clusters

The following subsections contain recommendations on how to set up an SGE cluster for efficiently running Ansys Electromagnetics serial and parallel jobs. These recommendations require the cluster administrator to make configuration changes.

Submitting Exclusive Jobs

Consumable Memory Limits

Serial Jobs in SGE

Parallel Jobs in SGE

Using Multithreading with Parallel Jobs

Submitting Exclusive Jobs

In many cases, clusters are used to run "large" Ansys Electromagnetics batch jobs. That is, these are jobs that may require a large quantity of resources, such as processors, memory, disk space, or run time. One way to ensure that the resources needed by the batch job are available to the job is to run the job in an "exclusive" mode. That is, any host running the job is not available for use by any other jobs. There is no SGE built in mechanism for specifying that a job is "exclusive". SGE is extensible, and it is not difficult to configure the cluster to allow exclusive jobs. The steps below show one way to do this. This example requires SGE 6.2u3 or later. Note that specifying a job as "exclusive" may delay the start of the job if there are not enough hosts available to run the job exclusively.

1. Use the command qconf -mcto add a new complex to the table of complexes. Recommended attributes are:
   - name : exclusive
   - shortcut : excl
   - type : BOOL
   - relop : EXCL

- requestable : YES
- consumable : YES
- default : 0
- urgency : 0

2. Set the value of "exclusive" to TRUE for each execution host using the command qconf -me hostname, where hostname is the name of the host. The values of all host configuration parameters may be displayed using the command qconf -se hostname. The "complex_ values" line should look similar to:

   complex_values exclusive=TRUE, but other values may also be included.

3. When submitting a job, the job will be "exclusive" if the value "excl" is included in the resource list specified by the qsub -l option. If the resource list does not include "excl" then the job will not be exclusive, and other jobs may run on the same host or hosts as this job.
4. Example qsub command line for exclusive serial job:

   ```
   qsub -b y -l excl /opt/AnsysEM/v231/ansysedt -ng
   -BatchSolve ~/projects/OptimTee.aedt.
   ```

   Although serial jobs use only one slot, no other jobs will run on the host where this job is running, even if additional slots are present.

5. Example qsub command line for exclusive parallel job using eight engines, each using a single thread of execution:

   ```
   qsub -b y -l excl -pe pe1 8 /opt/AnsysEM/v231/ansysedt -ng -
   BatchSolve -Distributed -machinelist num=8
   ~/projects/OptimTee.aedt
   ```

   None of the hosts used for this job will be allowed to run other jobs while this job is running.

Consumable Memory Limits

SGE contains several built-in complexes related to memory, including mem_total, for example, but none of these are "consumable". If a job is submitted with resource list including one of these non-consumable memory complexes (such as mem_total), then the job will run on a host or hosts only if sufficient memory is available. If a second job is submitted, the memory request for the second job is compared to the original total when determining if the job may run on a host. This may result in both jobs running out of memory. For example, if host A has mem_total=16G of memory, and two jobs are submitting with option "-l mt=16G", then both jobs could run on host A, if sufficient slots are available on host A.

SGE allows complexes to be "consumable" to avoid this type of problem. If a complex is consumable and a job requests x amount of the complex in the -l resource list, then the available amount of the resource is decreased by x for subsequent jobs. For the same example as above, if the mem_total complex was consumable, then the first job would run on host A. This would decrease the available mem_total from 16G to 16G-16G = 0. The second job could not run on host A because there is no memory available for this job.

The steps below show how to set up a consumable resource called physical_memory to accomplish the same thing. We do not recommend changing the behavior of the built-in

complexes (such as mem_total) because other scripts may expect normal behavior of the built-in complexes.

1. Use the command qconf -mcto add a new complex to the table of complexes. Recommended attributes are:
   - name : physical_memory
   - shortcut : phys_mem
   - type : MEMORY
   - relop : <=
   - requestable : YES
   - consumable : YES
   - default : 0
   - urgency : 0
2. Set the value of "physical_memory" to an appropriate value for each execution host using the command qconf -me hostname, where hostname is the name of the host. The appropriate value is the actual physical memory on each host. Because the type is MEMORY, the K, M, and G suffixes may be used to represent kilobytes, megabytes and gigabytes. The values of all host configuration parameters may be displayed using the command qconf -se hostname. The "complex_values" line should look similar to:

   complex_values physical_memory=16G,

   but other values may also be included, and the memory value should be appropriate for the host.
3. When submitting a job, the physical memory requirement per slot may be specified in the resource list as follows: -l phys_mem=mem_needed. The number of slots assigned to the job on a specific host will be limited by the number of slots available on the host, and also by the physical_memory available on the host.

Serial Jobs in SGE

If a serial job is submitted with the option -l phys_mem=mem_neededincluded, then the job may only run on a host in which the remaining physical_memory is equal to or greater than the mem_ neededvalue.

Example 1: Host A has physical_memory=16G, and host B has physical_memory=8G. If mem_ neededis 8G, the job may run on either host A or host B. If mem_neededis 16G, then the job may only run on host A.

Example 2: Host A has physical_memory=16G, and host B has physical_memory=8G. Job 1 is already running on host A, and it was submitted with option -l phys_mem=8G. If job 2 is submitted with option -l phys_mem=16G, then job 2 cannot start until job 1 finishes, because only host A has 16GB of physical_memory. If job 2 is submitted with option -l phys_mem=8G, then job 2 may start immediately, and run on either host A or host B, because both hosts have 8G of physical_memory remaining.

Parallel Jobs in SGE

Because the consumable setting for physical_memory is YES (and not JOB), each slot of the job requires a physical_memory of mem_needed. The number of slots on a host assigned to the job is limited by the number of available slots on the host. It is also limited by the physical_memory available on the host; the number of slots assigned to the job cannot exceed the available physical_memory on the host divided by the mem_neededspecification.

Example 1: Execution host A and execution host B both have 4 slots per host (configured in the queue associated with the parallel environment). Host A has physical_memory=16G and host B has physical_memory=8G (shown by commands qconf -se Aand qconf -se B). If a job is submitted that requires 6 slots and 4G per slot, it will be able to run, with 4 slots on host A and 2 slots on host B. The qsub command might look like: qsub -l phys_mem=4G -pe pe_name 6 command args

Example 2: Same as example 1, except that 7 slots are requested. In this case, the job will never run. Although there are 8 slots available on hosts A and B, only two of the slots on host B are usable by this job because it only has physical_memory of 8G. With only 6 slots total available to this job (4 on host A and 2 on host B), the job can not start. In this case the command might look like: qsub -l phys_mem=4G -pe pe_name 7 command args

Using Multithreading with Parallel Jobs

For large jobs it may be useful to combine multiprocessing with distributed processing. Distributed processing refers to starting multiple processes, in which each process performs a portion of the analysis. These processes may run on the same host or on different hosts. The number of processes running at the same time is known as the number of "analysis engines". Multiprocessing refers to using multiple threads within a single process to decrease the run time of the process. Multiprocessing may also be called multi-threaded processing.

As a concrete example of combining multiprocessing with distributed processing, an analysis could run with four engines, where each engine uses two threads. In order to distribute the processing load so that no processor is overloaded, one slot is generally allocated per thread, so 8 slots would be needed for this example (4 engines * 2 threads per engine = 8 threads). The four engines could all run on a single host, or they could be distributed across 2, 3 or 4 hosts, depending on available slots. Each engine represents a single process, so the two slots for each engine must be allocated on the same host.

This section describes how to set up an SGE cluster so that a specified number of slots per host may be requested when a job is submitted. This procedure will require the cluster administrator privileges. This capability may be used to submit parallel jobs in which one engine runs on each host, and the number of slots per host matches the number of threads used by each engine.

1. Let n be the largest number of slots available on any host used for the jobs. Create a separate parallel environment for each value of the number of slots per host from 1 to n. For example, pe_sph1 is a parallel environment in which one slot is allocated to the job per host, pe_sph2 is a parallel environment in which two slots are allocated to the job per host, etc. The command qconf -ap pe_namemay be used to create each new parallel environment. The allocation_rule parameter should be set to the number of slots per host, an integer from 1 to n. The control_slaves parameter should be set to TRUE, as described above. The slots parameter should be set to the maximum number of slots managed by this parallel_ environment, which is typically set to a large number, such as 999. The other parameters

should be set to values appropriate for the cluster. For example, the pe_sph2 parallel environment might have the following parameters:

- pe_name : pe_sph2
- slots : 999
- user_lists : NONE
- xuser_lists : NONE
- start_proc_args : /bin/true
- stop_proc_args : /bin/true
- allocation_rule : 2
- control_slaves : TRUE
- job_is_first_task : FALSE
- urgency_slots : min
- accounting_summary : TRUE

2. When submitting a job, use the parallel environment where the slots per host matches the number of threads per engine.

The batchoptions setting -machinelist num=n is required. This should be set to match the number of slots per host. With any analysis, a portion of the analysis may not be distributed across multiple engines.

Example qsub command line for running distributed processing with four engines and multiprocessing with two threads per engine:

```
qsub -V -b y -pe pe_sph2 8 "/opt/AnsysEM/v231/ansysedt
-ng -BatchSolve -Distributed -machinelist num=4 -batchoptions
"projects/OptimTee.aedt"
```

The -Voption indicates that the all environment variables in the submission environment should be copied to the job environment.

- The -b y option indicates that Maxwell is launched directly from the command line, instead of using a script.
- The -pe sph2 8 command_line option indicates that this is a parallel job running under the pe_sph2parallel environment so that two slots are allocated to this job from each host, and that 8 slots in total are allocated to this parallel job.
- The -Distributed option indicates that this is a DSO job, so that multiple engines will be started. The -Distributed option may now have additional options, such as includetypes=xxx, excludetypes=xxx, maxlevels=n, and numlevel1=n, where n indicates and integer, and xxx indicates a list of distribution types or "default".
- The -machinelist num=4option indicates that a total of four engines will be started.
- The entire Maxwell command is in double quotes, and the double quotes enclosing the -batchoptions value are escaped. Each of these double quotes is replaced by the sequence "\"".

Related Topics

Integration with Sun Grid Engine (SGE)

Monitoring Ansys Electromagnetics SGE Batch Jobs

Issue with qrsh (SGE)

Ansys Electromagnetics parallel batch jobs use the SGE **qrsh** command to launch engine processes on remote hosts. If the **qrsh** command is not working correctly, then the parallel job is unable to launch engine processes on remote hosts. If this problem occurs, the batch log for the job typically includes one or more error messages indicating that a COM engine was unable to be started on a remote host. If this occurs, the user or cluster administrator should verify that the SGE qrsh command is working correctly, and correct the problem if the SGE **qrsh** command is not working correctly.

The **qrsh** command may be tested by running a simple command on a specified host, such as **qrsh -l hostname=host1 hostname** or **qrsh -l hostname=host1 ls /tmp**, where host1 is the remote host name. The first test should simply echo back the hostname of the remote machine. The second test should list the contents of the /tmp directory on the remote machine.

The failures of the SGE qrsh command are associated with the following global sge configuration parameters, listed below with values that may cause the failures:

```
qrsh_command /usr/bin/ssh -t
rsh_command /usr/bin/ssh -t
rlogin_command /usr/bin/ssh -t
```

If these parameter settings are removed, then the SGE built-in mechanisms are used for qrsh, rsh, and rlogin. No problems with the built-in versions have been reported. The SGE qconf -sconf global command may be used to view these parameter settings. The SGE qconf -mconf global command may be used to modify or remove these parameter settings.

Issue with MainWin Core Services for SGE

By default, SGE creates a temporary directory for each SGE batch job, and deletes this temporary directory and its contents when the job finishes. SGE sets the TMP and TMPDIR environment variables of the job environment to point to this temporary directory. Ansys Electromagnetics desktop software starts the MainWin Core Services on startup, if they are not already running. After the Ansys Electromagnetics desktop software finishes, the MainWin Core Services time out and automatically shut down. The MainWin Core Services use the TMP and/or TMPDIR directories to store temporary data. If this temporary data is removed before the services shut down, then the services do not shut down automatically. Normally, SGE will remove the temporary directory and its contents before the services time out. The result is that these extraneous service processes run forever. If this problem occurs, each Ansys Electromagnetics batch job starts an additional set of these services that never shut down. This can result in an excessive number of processes running on the host where the Ansys Electromagnetics desktop is started. The names of the service processes are:

- watchdog
- regss
- mwrpcss

Workaround for Issue with MainWin Core Services

One way to avoid this problem is to modify the environment in which the Ansys Electromagnetics desktop runs so that the TMP and TMPDIR environment variables do not point to the directory which will be immediately removed by SGE when the job finishes. This can be done by copying the value of the TMPDIR environment variable to the ANS_SGE_TMPDIR environment variable, and unsetting the TMPDIR and TMP environment variables. The services ignore the ANS_SGE_TMPDIR environment variable, but if this variable is set, then it will be used as the temporary directory for the rest of the Ansys Electromagnetics software.

Here is an example bash wrapper script that may be used to work around this issue. In this example, the product is maxwell, but the same approach will work for any Ansys Electromagnetics product. In this example, the script is named sge_maxwell and is in the AnsysEM software installation directory. When an Ansys Electromagnetics desktop job is submitted to the SGE scheduler, the script (sge_maxwell, in this example) should be submitted instead of maxwell. The script will modify the environment, as needed, then start maxwell. When the analysis finishes, the script returns the exit status of maxwell.

An alternative is to place the script in an arbitrary directory, and modify the script to include an absolute path to the product (maxwell in this example).

Script contents:

```
#! /bin/bash

# This script will not correctly process arguments containing
# spaces or other characters special to the shell.

# Create maxwell command line
# In this example, sge_maxwell and maxwell are in the same
directory
# An alternative is to use an absolute path for the maxwell
command
cmd0=$0
cmd="${cmd0/%sge_maxwell/maxwell} $@"

# Fix environment variables
export ANS_SGE_TMPDIR=${TMPDIR}
unset TMPDIR
unset TMP
```

```
# Run the maxwell command and return the exit status
${cmd}
exit $?
```

Related Topics

Integration with Sun Grid Engine (SGE)
Monitoring Ansys Electromagnetics SGE Batch Jobs
Ansys Electromagnetics Desktop -monitor Command Line Option for SGE
Example SGE qsub Command Lines
Issue with MainWin Core Services for SGE
Recommended Practices for SGE Clusters

Integration with Platform's Load Sharing Facility (LSF)

The Load Sharing Facility (LSF) scheduler is only supported on Linux. Jobs may be submitted in any of the following ways:

- Job Submission GUI
- Using LSF commands (qsub, etc.)

See the Ansys Electromagnetics Suite 2023 R1 Unix/Linux Installation Guide for additional information on supported schedulers.

> **Note:**
>
> If a temp directory is setup by the LSF cluster administrator, analysis engines use this temp directory, overriding the setting in the Ansys EM product.

LSF Job Management

You can use Ansys Electronics Desktop to submit batch jobs to LSF and monitor those jobs.

This involves the following steps:

1. Use **Tools** > **Job Management** > **Select Scheduler** to select LSF as the scheduler.
2. Use **Tools** > **Job Management** > **Submit Job** to submit a batch job to LSF.

   LSF-specific Settings:

   - On the **Compute Resources** tab, click the **Resource Selection Parameters** ellipses button (**...**) to specify the following:
     - **Queue** – a drop-down menu lets you select Normal, chkpn_rerun_queue, idle, license, night, normal_allow_excl, owners, priorityu, or short.

- **MinCoresPerNode** – the minimum number of cores allowed on a node to be eligible for selection; translates to bsub -R select[ncpus>=N]
- **MaxCoresPerNode** – the maximum number of cores allowed on a node to be eligible for selection; translates to bsub -R select[ncpus<=N]
- **MinMemoryPerNode** – the minimum amount of physical memory (specified in integer GigaBytes) allowed on a node to be eligible for selection; translates to bsub -R select[maxmem>=M]
- **MaxMemoryPerNode** – the maximum amount of physical memory (specified in integer GigaBytes) allowed on a node to be eligible for selection; translates to bsub -R select[maxmem<=M]

If you do not specify parameters, the scheduler does so.

- For LSF, the only non-automatic method of **Resource Selection** is **Number of Tasks and Cores**. You can specify the number of tasks, whether they are for exclusive use by the job, cores per distributed task, a limit number of tasks per node, and RAM limit as percent.

3. Use **Tools** > **Job Management** > **Monitor Job** to monitor the job's progress.

LSF Job Submission Guidelines

Before submitting an LSF job, ensure the following are true:

- The project is available in a shared drive that is accessible to all machines in the cluster.
- The project is available using the same path on all machines of cluster.
- There is sufficient space in the project directory and temp directories.
- There is sufficient memory per engine.
- The number of compute resources (Distributed Analysis machines and Multi Processing cores) will achieve the desired scale factor and effective resource utilization.

Integration of Ansys EM Products with LSF

With LSF you do not need to set 'Distributed Analysis Machines' or 'Remote Machine' options. Instead, you submit a job to LSF, requesting the appropriate resources for the job (number of processors, memory per processor, etc.).

For example:

```
bsub -n 1 ansysedt.exe -Batchsolve -ng -local -machinelist num=1
OptimTee.aedt
bsub -n 4 ansysedt.exe -Batchsolve -ng -Distributed -machinelist
num=4 OptimTee.adsn
```

The job is queued by LSF until the requested resources are available. Upon resource availability, LSF starts Electronics Desktop with the specified command line on one of the allocated machines. During analysis, Electronics Desktop dynamically obtains the allocated 'Distributed Analysis

Machines' from LSF. Electronics Desktop interfaces with LSF to launch engines on remote machines without going through Ansys RSM.

Installing Ansys EM Tools on LSF Cluster

The LSF scheduler is supported on Linux only.

lsf.conf should contain this line:

```
LSF_UNIT_FOR_LIMITS=MB
```

The administrator should have this line in the lsb.params file to ensure that memory reservations are per-slot (per-core):

```
RESOURCE_RESERVE_PER_SLOT=Y
```

Jobs may be submitted in any of the following ways:

- Using LSF commands (bsub, etc.)
- Using the generic scheduler GUI in local mode
- Using the generic scheduler GUI in service mode

See the Ansys Electromagnetics Suite 2023 R1 Unix/Linux Installation Guide for additional information on supported schedulers.

Setup:

1. Install on a single node, on a shared drive.
2. Setup 'temp directory' to a path that is same on all nodes (for example, /tmp).
3. Ensure that the product is available using the same path on all nodes.

Permissions:

- All users of the cluster should have read/write permissions to temp directory.
- All users should have read/execute permissions to installation directory.
- Turn OFF firewall between cluster nodes.

When a desktop scheduler GUI is run the same node as the job submission node, no other configuration is necessary; installation is sufficient. Select the scheduler through the Electronics Desktop GUI. Ensure that scheduler commands are available in the path before you launch Electronics Desktop.

Note:

There is no need to install RSM unless you are using the scheduler GUI on a post-processing node that is different from the job submission node. In this case, RSM must be configured with the scheduler type and path.

A post-processing node is a node in the cluster that can run Electronics Desktop in graphical mode. A job submission node is a node in the cluster in which job submission commands are available.

Per-slot Resource Reservation

Set the cluster for per-slot resource allocation if the automatic cores and RAM resource selection method is to be used. You can check the cluster to see if per-slot resource allocation is configured by using the "bparams -a" command. Search the output for "RESOURCE_RESERVE_PER_ SLOT" to determine the setting (either "Y" or "N"). If set to "N" then consult the LSF administration guide on how to change this to "Y".

Scenario 1: The post-processing node and job-submission node roles are served by distinct machines.

In this case, perform the following configuration:

The job-submission node should be configured to run RSM service, which serves as a proxy to scheduler. The RSM Service should be running as 'root' in order to facilitate jobs running using the credentials of the job's owner. A **configuration file in the RSM installation folder should be edited** to specify information regarding the scheduler that manages jobs on this cluster. A block labeled 'Scheduler' must be included within the 'AnsoftCOMDaemon' block. This block contains two string entries:

- SchedulerName: this contains the unique part of the scheduler proxy library name
- ConfigString: this contains a scheduler specific configuration string

The case of the SchedulerName string is significant on Linux because Linux file names are case sensitive. In Ansys Electromagnetics Suite 2023 R1, possible scheduler names are: lsf and sge. The ConfigString entry is a scheduler specific configuration string, described below.

In addition, the AnsoftRSMService must be started with appropriate environment variables set. Generally, the environment variables must be set the same as they would be set for using the scheduler via command lines.

LSF Details

For the LSF scheduler proxy library, the ConfigString entry in the ansoftrsmservice.cfg configuration file is ignored. It may be empty or omitted entirely.

The AnsoftRSMService must be started with the environment set as it would be set for submitting jobs to the LSF cluster.

- For Linux, the cshrc.lsf or the profile.lsf file may be sourced to set up the environment, depending on the shell.

Example ansoftrsmservice.cfg configuration file:

```
$begin 'AnsoftCOMDaemon'
$begin 'Managed COM Servers'
$end 'Managed COM Servers'
$begin 'Scheduler'
'SchedulerName'='lsf'
'ConfigString'=''
$end 'Scheduler'
$end 'AnsoftCOMDaemon'
```

Scenario 2: The post-processing node and job-submission node roles are served by the same machine.

The **Select Scheduler...** command (as described in the Job Management User Interface for LSF section) is used to gather details about the scheduler. In this case, the Desktop process should be started in an environment suitable for submitting jobs to the scheduler. See below for details.

The environment should be configured so that the following LSF environment variables are set appropriately for the LSF cluster in use: LSF_BINDIR, LSF_SERVERDIR, LSF_LIBDIR, and LSF_ENVDIR. In addition, the following LSF commands should be found in the LSF_BINDIR directory: "bsub", "bjobs", "bkill", "lsid", "lsrun", "lshosts", "bmgroup", "bparams" and "bqueues".

Using the bsub Command to Submit Batch Jobs

The LSF bsub command may be used to submit jobs. The typical command format is:

`bsub` *`bsub_args ansys_exe ansys_args`*

where:

- *bsub_args* are the options of the `bsub` command,
- *ansysEM_exe* is the pathname of the Ansys Electromagnetics desktop executable to launch, and
- *ansys_args* are the arguments to the Ansys Electromagnetics desktop executable.

bsub Arguments

The **bsub** command has a large number of options that may be used to control the submission process. Only a few options that are often used with Ansys Electromagnetics jobs are listed here. The following options may be used to submit serial or parallel LSF jobs.

-n*`min_proc, max_proc`* or **-n***`min_proc`*

Submits a parallel job, specifying the number of processors (or slots) required for the job. Here, *min_proc* is the minimum number of processors, and *max_proc* is the maximum number of processors. If no maximum is specified, then exactly *min_proc* processors are requested. If

PARALLEL_SCHED_BY_SLOT=Y in lsb.params, this option specifies the number of slots required to run the job, not the number of processors. If the **-n** command line option is not specified, then the job is submitted as a serial batch job.

-R "span[ptile=n]"

There are many ways to use the -R "res_req" option to the bsub command. We only cover -R "span[ptile=n]" here, because this option is very useful for Ansys Electromagnetics jobs. When this option is specified, the LSF scheduler will allocate n processors (or slots) on each host to this job, even if more processors are available on the host.

-x

All hosts running this job operate in exclusive execution mode. The job will only run on a host having no other jobs running on that host. No other batch jobs will be started on a host while this job is running on that host.

See LSF documentation for a complete list of options for the **bsub** command.

> **Important:**
>
> If the Ansys EM tool executable pathname (*ansys_exe*) or any of the arguments of the Ansys tool command (*anssys_args*) contain characters which are interpreted by the command shell, these special characters must be properly quoted to ensure that the correct command is launched by LSF. A similar problem may occur if any of the *ansoft_args* require single quote, double quote or space characters. Note that the Ansys Electronics Desktop command is processed by the shell twice: once when the **bsub** command is processed, and again when the job starts.

Example LSF bsub Command Lines (Linux Only)

> **Note:**
>
> The following examples use HFSS as the product, but similar command lines will work for all Ansys EM products.

Serial Job

```
bsub -n 1 /Program Files/AnsysEM/v231/Win64/ansysedt -ng
-BatchSolve -machinelist num=4 ~/projects/OptimTee.aedt
```

The -n 1 option indicates that this job runs on one core.

Serial Job Requiring a Minimum of 4GB

```
bsub -n 1 -R "select[mem>4000]"
/Program Files/AnsysEM/v231/Win64/ansysedt -ng
-BatchSolve -machinelist num=4 ~/projects/OptimTee.aedt
```

The -R "select[mem>4000]" option indicates that this needs a minimum of 4 GB memory.

Multi-processing Job using 4 Cores

```
bsub -n 4 -R "span[ptile=4]"
"/Program Files/AnsysEM/v231/Win64/ansysedt -ng -BatchSolve
-batchoptions -machinelist num=4 ~/projects/OptimTee.aedt"
```

- The -R "span[ptile=4]" option indicates that the four cores need to be on the same machine.
- The -batchoptions option indicates that HFSS should use four cores for multi-processing.
- The entire hfss command is in double quotes, and the double quotes enclosing the -batchoptions value are escaped. Each of these double quotes is replaced by the sequence "\"".

Distributed Processing Job using 4 Engines

```
bsub -n 4 /Program Files/AnsysEM/v231/Win64/ansysedt -ng -
BatchSolve -Distributed ~/projects/OptimTee.aedt
```

- The -n 4 option indicates that the four cores are needed for the job.
- The -Distributed option indicates that this is a DSO job, so that multiple engines will be started. Because 4 cores are allocated to the job, the job will run 4 engines. The -Distributed option can have additional options, such as includetypes=xxx, excludetyeps=xxx, maxlevels=n, and numlevel1=n, where n indicates and integer, and xxx indicates a list of distribution types or "default".

Distributed Processing and Multi-processing Job using 4 Cores, with 2 Cores for Multi-processing

```
bsub -n 4 -R "span[ptile=2]" ~/projects/OptimTee.csh
```

Shell Script (~/projects/OptimTee.csh):

If a command is included in the **bsub** command line, the entire command will be processed by the command shell two times. The command is processed when the **bsub** command is processed by the shell and is processed again when the command is started by the scheduler. This example shows how to use a shell script so that the command line will be processed only once. The command is placed in the shell script, and then the shell script pathname is placed in the **bsub** command line. Then, the command is only processed by the command processor when the job is started. When using this approach, the shell script should be accessible from all of the cluster hosts.

```
#!/bin/csh
/Program Files/AnsysEM/v231/Win64/ansysedt -ng -BatchSolve
-Distributed -machinelist num=2 -batchoptions
~/projects/OptimTee.aedt
```

- The **-n 4** option indicates that the four cores are needed for the job.
- The **-R "span[ptile=2]"** option indicates that the cores must be allocated in groups of two cores on the same machine.
- The **-machinelist num=2** option indicates that this is a DSO job and that a total of two engines will be started.
- The **hfss** command is placed in the shell script (~/projects/OptimTee.csh). In the **bsub** command line, the **hfss** command is replaced by the shell script pathname.
-

Monitoring LSF Batch Jobs

You can monitor jobs through the Electronics Desktop user interface, or through the command line.

The suggestions below are for batch jobs run under LSF.

Ansys Electronics Desktop -monitor Command Line Option

The -monitor command line option enables batch job output to the standard output and standard error streams. The warning, info, and progress messages are sent to the standard output stream. The error and fatal messages are sent to the standard error stream.

LSF bpeek Command

The LSF bpeek command may be used to monitor job progress. The command **pbeek [-f]** *job_id* displays the standard output and standard error produced by the job with id *job_id* from the job start to the current time (the time when the command is executed). This command is only valid for jobs that have not yet finished. When used with the **-f** option on Linux, the output of the job is displayed using the command tail **-f**, so that ongoing progress may be monitored.

In order to display messages to standard output and standard error, specify the **-monitor** command line option on the Ansys EM tool command line. Then, these messages can be seen using the LSF bpeek command.

Terminating LSF Batch Jobs

To cancel or terminate an Ansys EM LSF batch job, we recommend using the job monitoring UI to terminate jobs cleanly, rather than using the bkill commands. Using this approach will allow the batch job to shut down in an orderly fashion.

Using the LSF **bkill** command without the **-s SIGTERM** option or simply terminating the job processes may cause some of the following problems:

- Some engine processes are not shut down and continue to run.
- LSF job is not fully removed.

- Project .lock file is not removed.
- MainWin core service processes (watchdog, mwrpcss and/or regss) are not stopped.

Some of these may interfere with submission of additional LSF batch jobs. For example, it may be necessary to manually remove the project lock file to submit another batch job for the same project. MainWin core service processes may also interfere with starting subsequent Ansoft batch jobs. Normally, these processes should timeout and end 15 seconds after the Ansys Electromagnetics product shuts down. Any MainWin core service processes (watchdog, mwrpcss and/or regss) that continue to run for more than 15 seconds after the product has stopped may be hung. The hung processes may need to be manually killed, after ensuring that these processes are associated with an Ansys EM job that has finished or terminated.

Stop a job cleanly - ensures that the results obtained until now are preserved:

```
bkill -s TERM <jobid>.
```

Stop an job abruptly - results are most likely lost. You have to manually remove the project lock file:

```
bkill <jobid>
```

LSF Known Issues and Workarounds

The following are known issues. Workarounds are noted when available:

- Desktop or remote machine cannot have multiple IP addresses. This is unsupported.
- Core dump files may appear when a job has finished running. Results are still computed correctly. Workaround: Limit size of core dumps to 0 through the following job submit option: `bsub -C 0 -n <number-of-cores> -q <queue-name>`
- Firewall should be turned off on all machines in the cluster.
- Sometimes LSF ends a job (for example, a job may be preempted due to a high priority job). This may result in the presence of a .lock file in the project directory. You must manually delete the .lock file before continuing with further analysis.
- When an LSF job is ended, MainWin services (watchdog, regss, and mwrcpss) could keep running. The result is that later jobs cannot start on the machine. The fix is to end these processes before starting a new job.
- Analysis fails abruptly when running out of resources (cpu/memory/disk). Ensure sufficient resources are provided.

LSF Troubleshooting

The following are general troubleshooting steps:

1. Ensure the LSF `lsrun` command is enabled.
2. Look for user errors.

   For example:

- Are the executable path and project path correct and complete?
- Are there sufficient resources (CPU/Memory/Disk) allocated to the job?
- Is the project available on the execution host?
- Does the job submitter have read/writer permissions on the project directory and read/execute permissions on the installation directory?
- Is the project locked?

3. Determine whether this is a standalone product issue.
   - Run Electronics Desktop on the machine outside of the scheduler and see if it opens and analyzes.
4. Examine outputs and logs.
   - Output of the LSF batch job. Obtain this using LSF commands: "`bacct -l <jobid>`"
   - Batch log (typically <projectname>.log, located in the project directory.
5. Enable additional debug logs using the steps below.

   In the job submission window, set the following environment variables:
   - ANSOFT_DEBUG_MODE = 1
   - ANSOFT_DEBUG_LOG = <*path to directory accessible by all machines in the cluster*>
   - ANSOFT_DEBUG_LOG_SEPARATE = 1
   - ANSOFT_LSF_LOG = <*path to a specific .log file in the directory set under ANSOFT_DEBUG_LOG*>
6. For each pair of machines between which remote analysis fails, run `ping remote-machine` and note the output.
7. For each machine in the network, dump network interfaces (for example, run `ifconfig -a`) and note the output.

Integration with PBS (Portable Batch System)

The PBS Pro and PBS/Torque schedulers are only supported on Linux. Jobs may be submitted in any of the following ways :

- Job Submission GUI
- Using PBS commands (qsub, etc.) or the PBS gui (xpbs)

See the Ansys Electromagnetics Suite 2023 R1 Unix/Linux Installation Guide for additional information on supported schedulers.

PBS Job Management

You can use Ansys Electronics Desktop to submit batch jobs to PBS and monitor those jobs.

This involves the following steps:

1. Use **Tools** > **Job Management** > **Select Scheduler** to select PBS as the scheduler.
2. Use **Tools** > **Job Management** > **Submit Job** to submit a batch job to PBS.

   PBS-specific Settings:

   - On the **Compute Resources** tab, click the **Resource Selection Parameters** ellipses button (**...**) to specify either the **Queue** or **QueueAtServer** parameter.

     For **Queue**, you may select a queue for the job from the list of queues configured for the default server. Only queues that are enabled and that do not have the from_ route_only attribute set to true are listed.

     For **QueueAtServer**, you may specify a queue at the default server or at another server by entering text into this field in one of three formats:

     - queue_name
     - @server_name
     - queue_name@server_name

     The queue_name format submits with the name queue_name. The other formats submit with the name server_name. The *destination* value of the -q *destination* option on the qsub command line is the user-specified string. *This string will not be validated by the scheduler proxy library*.

     Either **Queue** or **QueueAtServer** may be specified, but both may not be specified. If neither **Queue** nor **QueueAtServer** is specified, the job is submitted to the default queue at the default server.
3. Use **Tools** > **Job Management** > **Monitor Job** to monitor the job's progress.

Related Topics

Selecting a Scheduler

Submitting a Job

Monitoring Jobs

Running Ansys Electronics Desktop from the Command Line

Command Line Information for HPC

Non Standard Installations for PBS

PBSPro

If the environment variable PBS_DEFAULT is set, the value of this environment variable will be used as the name of the default server, instead of obtaining the default server name from the PBSPro configuration file. The default pathname of the PBSPro configuration file is: `/etc/pbs.conf`. The environment variable PBS_CONF may be used to specify a different pathname for the PBSPro configuration file.

PBS/Torque

If the environment variable PBS_DEFAULT is set, the value of this environment variable will be used as the name of the default server, instead of obtaining the default server name from the PBS/Torque server file. The name of the PBS/Torque server file is server_name, and it is installed in the TORQUEHOME directory. By default, TORQUEHOME is `/var/spool/torque`. To specify a different TORQUEHOME directory, the environment variable ANSOFT_ TORQUEHOME should be set to the pathname of the desired directory.

Related Topics

Integration with PBS (Portable Batch System)

PBS Limitations

General Limitations

- There is no support for GPUs when submitting jobs via the GUI.
- Support for PBSPro and PBS/Torque is only available on Linux; Windows is not supported.
- Staging of input or output files is not supported for jobs submitted using the GUI.
- All jobs submitted via the GUI are independent jobs. Neither job dependencies nor job arrays are supported.
- If the user -specified server is not the default server, there is no check for sufficient resources before submitting the job.
- If the user specified server is not the default server, the limit on the number of tasks per node is ignored for both PBSPro and PBS/Torque. For PBS/Torque, only one task will be allocated for each node. For PBSPro, the scheduler may allocate any number of tasks to a node, provided that the node has sufficient cores for all of the tasks.
- For jobs submitted to a routing queue, the check for sufficient nodes and cores by the scheduler proxy only verifies that there are sufficient nodes and cores associated with the server.
- The queue attributes “resources_max” and “resources_min” are not checked when determining whether there are adequate resources to run the job.

PBSPro Limitations

- The PATH in the submission user’s default environment must include the directory containing the PBSPro commands.
- Failover is not supported.
- HPC Basic Profile Jobs are not supported.
- Globus vnodes are not supported. When checking for sufficient nodes and cores for the job, only nodes of type PBS are considered.
- For jobs submitted to an execution queue, the only vnode attribute used to determine if a vnode is available to the job is the “queue” attribute.

PBS/Torque Limitations

- When checking for sufficient nodes and cores for the job, only nodes of type cluster are considered.
- The “exclusive” check box has no effect for PBS/Torque.
- For PBS/Torque, even if a job is submitted to an execution queue at the default server, there is no check for sufficient nodes and cores available to the queue. All of the server’s execution nodes are assumed to be available for the job.
- For PBS/Torque, there are significant limitations when submitting a job in which the number of tasks and number of cores per task are specified. Unlike PBSPro, there is no capability to specify that the cores should be allocated in “chunks”. Instead, the submission command includes the number of groups of nodes and the number of processors per node (ppn) for each node in the group. To determine the size of each group and the ppn setting for each group, the server nodes are examined from largest number of cores to smallest. This may not be optimal because some of these nodes may not be usable by the queue specified for the job, or because the nodes with the largest number of cores may be busy. Similar issues could occur for PBSPro, but they should be less likely, because only the nodes usable by the queue are considered.

Submitting Jobs Via Ansoftrsmservice on a Different Host

Before starting the ansoftrsmsevice as a daemon on a job submission host, the 'Scheduler' section of the ansoftrsmservice.cfg must be specified.

This section contains two settings, 'SchedulerName', which must be set the string 'pbs', and 'ConfigString', which must be set to the pathname of the directory containing the PBSPro or PBS/Torque commands.

Here is an example ansoftrsmservice.cfg file, showing the format of this file and an example 'ConfigString' setting:

```
$begin 'AnsoftCOMDaemon'
$begin 'Managed COM Servers'
$end 'Managed COM Servers'
$begin 'Scheduler'
'SchedulerName'='pbs'
'ConfigString'='/share/pbs/default/bin'
$end 'Scheduler'
$end 'AnsoftCOMDaemon'
```

Related Topics:

Integration with PBS

Submitting Ansys EM PBS Batch Jobs

The PBS qsub command may be used to submit Ansys EM batch jobs. The typical command format is:

```
qsub qsub_args script
```

where:

- *qsub_args* are the options of the **qsub** command,
- *script* is the pathname of the job script.

The job script is a shell script containing the Ansys batch command or commands to be run. If a batch command line contains any characters that are special to the shell running the script, then these special characters should be quoted, as needed. The job script may also contain PBS directives on lines before the first executable line of the script. Any **qsub** options on the command line will take precedence over the PBS directives in the job script.

When a PBS batch job is started, the job script runs as the job user in a new shell. In this shell environment, the path must include the directory containing the PBS commands.

> **Note:**
>
> You should ensure that the PATH variable set in the shell startup script (e.g., .cshrc, .profile, .bashrc) includes the directory containing the PBS commands. For example:
>
> ```
> export PATH=/opt/pbs/default/bin:$PATH
> ```
>
> If the PATH variable is not set correctly, the job runs only locally, the batch log file shows the list of allocated hosts as empty, and the error file shows an error (sh: qstat: command not found.)

Further PBS directives need to be on top of the job script file. This is discussed in the PBS documentation.

Serial PBS Batch Jobs

In the PBS documentation, serial batch jobs are also called single-node jobs. In general, any job submitted without specifying the -l nodes=value command line argument, will run as a serial or single-node job.

See Monitoring PBS Batch Jobs for options that can facilitate monitoring of Ansys Electromagnetics batch jobs.

Parallel PBS Batch Jobs

In the PBS documentation, parallel batch jobs are also called multi-node jobs. When an Ansys Electromagnetics batch job is run as an PBS parallel job, the PBS scheduler will select the hosts for the distributed analysis job based on the qsub command line arguments, the PBS resource directives from the job script, and the status of the hosts when the job is run. The desktop process will be started on one of these hosts. The desktop process will obtain the list of hosts allocated to the job from the PBS scheduler, and start analysis processes on these hosts, as needed, using the PBS scheduler facilities. To run a PBS parallel job, the job must be submitted with a -l nodes=value **qsub** command line argument or with a -l nodes=value PBS directive in the job script.

Related Topics:

Monitoring PBS Batch Jobs

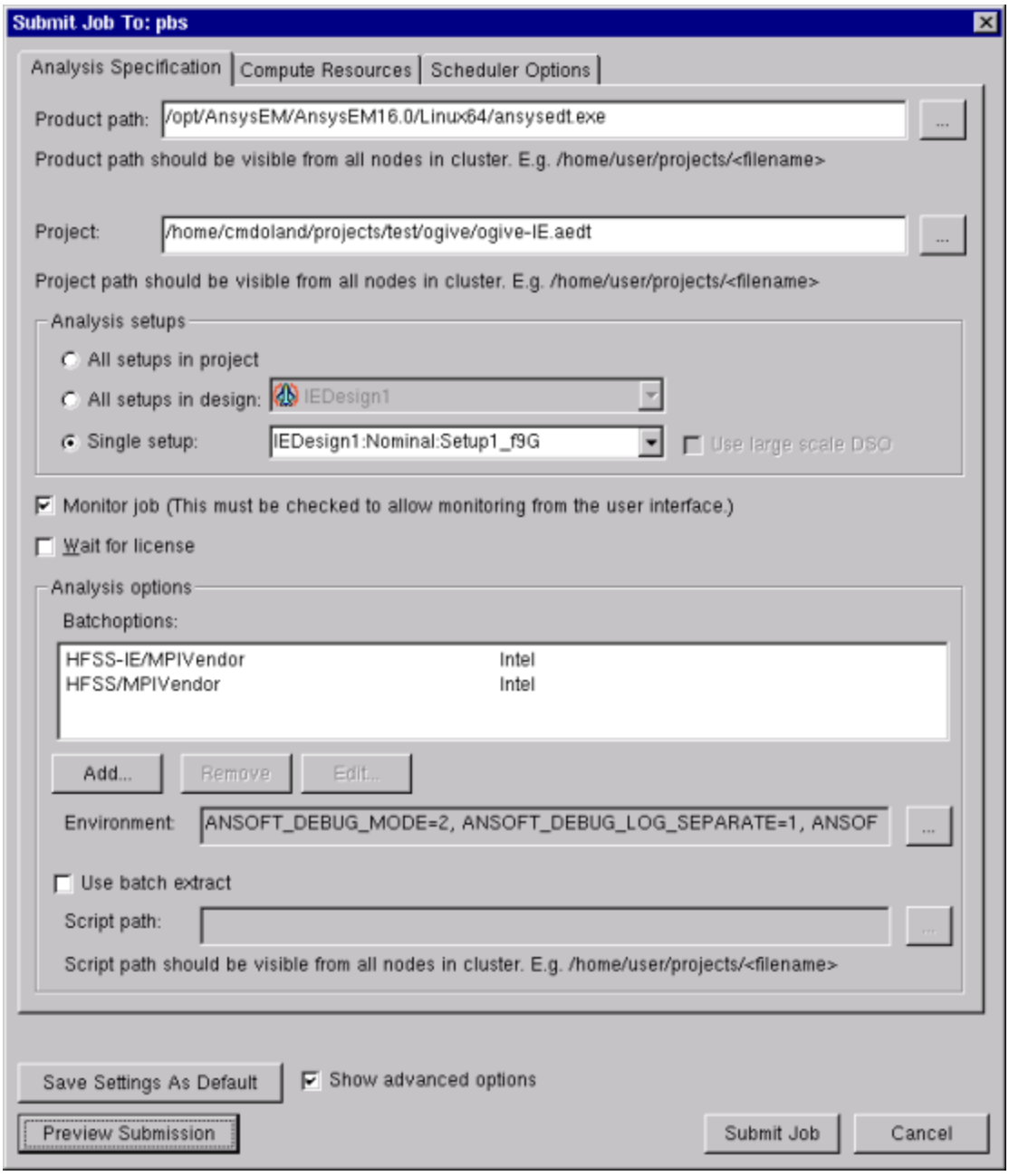
Example PBS qsub Command Lines

What a Scheduler Does

Job Management User Interface for PBS

After setting the job submission node, select **Tools>Job Management>Submit Job...** to open the **Submit Job To:** dialog box. This contains three tabs:

- **Analysis Specification**--specify the Product path, Project name, the setups, job monitoring, and analysis options such as batchoptions, or, for advanced users, Environment variables.
- **Compute Resources**--this tab can be populated either by automatic settings, by predefined Analysis Configuration, or specifying parameters in the fields for resource selection, for job parallelization and enabled forms of parallelization.
- **Scheduler Options**--contains fields for Job name and priority. The customization options shown by checking advanced are not used for Windows HPC.



In the **Analysis Specification** tab, enter the pathnames of the product path and of the project file in the “Project” edit box. These must be UNC paths that are accessible from each compute host used for Ansys Electromagnetics jobs. The submission user must have permission to write to the directory containing the project file. The Project can be an archive.

You can select which setups are analyzed in the Analyze Setups section of this dialog box. There are radio buttons to select:

- All setups in the project
- All setups in a specified design: you selects the design from the dropdown list

- Single setup:

If you specify multiple setups, they will be processed sequentially in the order displayed in the edit box.

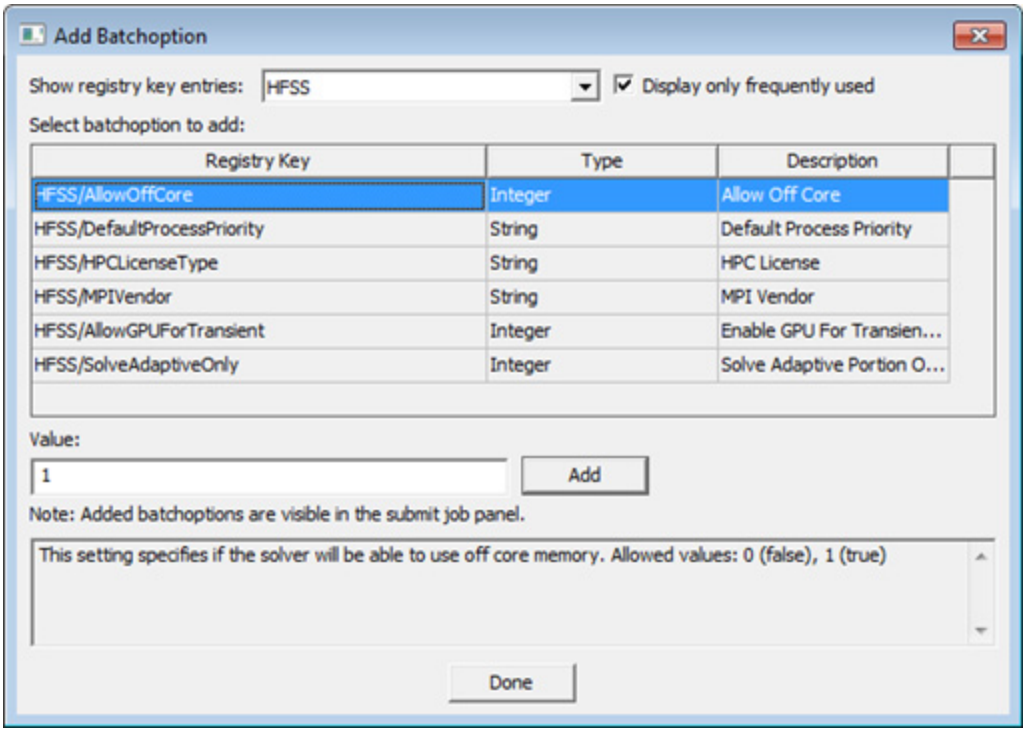
The Analysis options include:

- Monitor job. You must enable this option to monitor the job from the user interface.
- Wait for license- whether to wait until a license if available before starting a simulation.
- Batch options. You can optionally specify -Batchoptions in the text field. See detailed discussion of -Batch Options beginning under Running Maxwell from a Command Line.

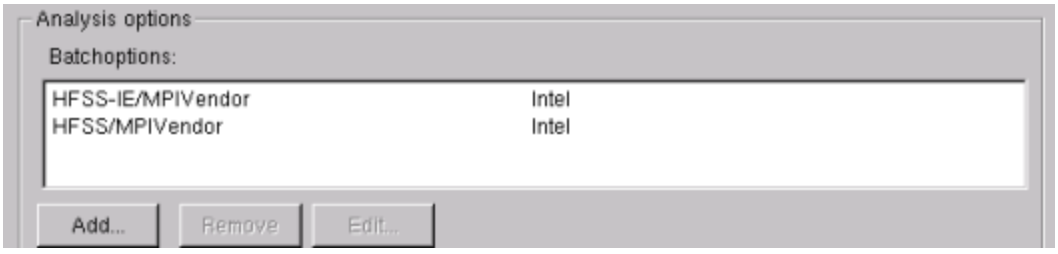
The **Add...** button opens the **Add Batchoption** dialog.

This dialog provides access to all -Batchoption commands. The drop down menu lets you select specific categories, and you can choose to display only frequently used commands. You can edit and remove any batch options you specify.

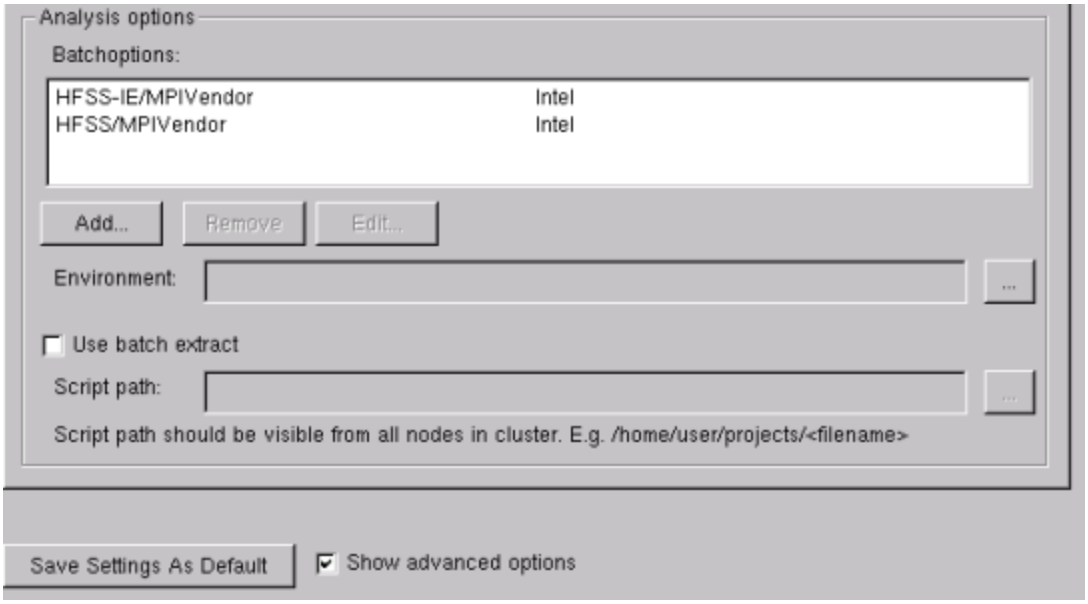
Select a Registry Key. in order to show the current Value for the type. The lower field explains the meaning of the Type Value.



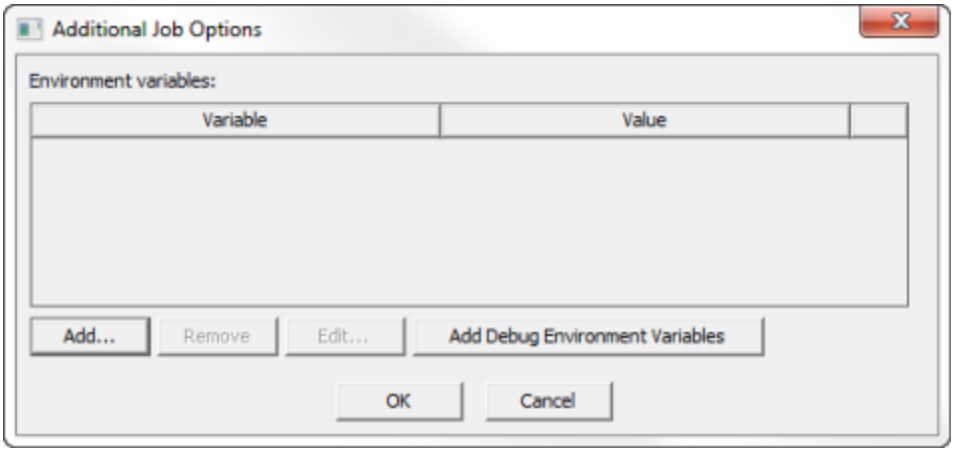
Any batchoptions for which you select **Add** will be visible in the **Submit Job** dialog.



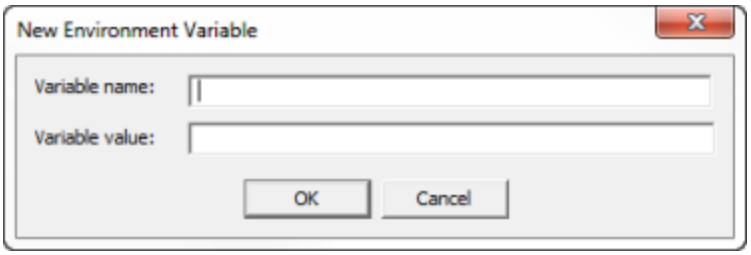
If you have the Show advanced options box checked in the **Submit Job** dialog box, the Environment field and the Use batch extract fields display.



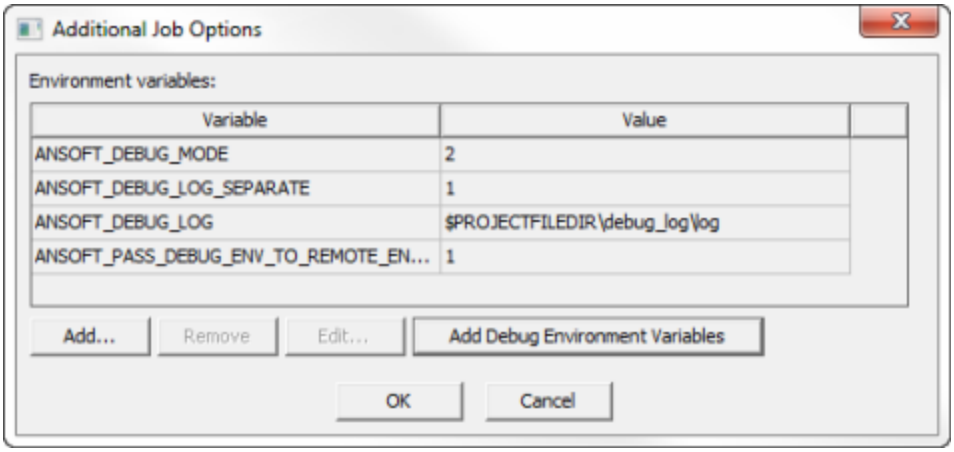
The Environment field lets you specify any Environment variables. Click the ellipsis button [...] to display the **Additional Job Options** dialog.



Click the **Add...** button to open the **New Environment Variable** dialog.



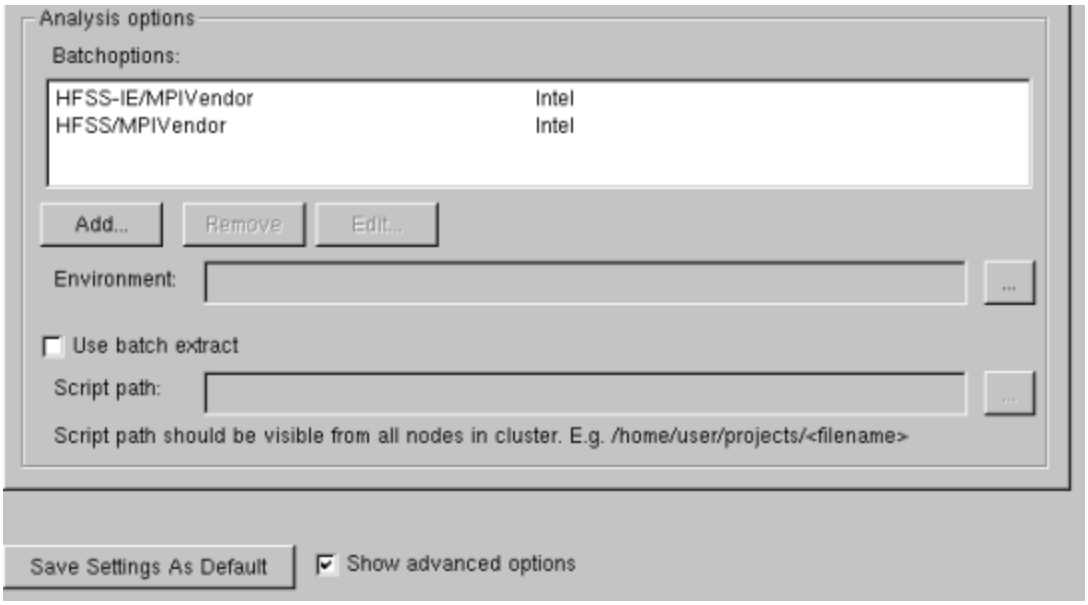
Here you can provide a Variable name and Variable value. Click OK to display the Variable in the Additional Job Options dialog box. Select a Variable to enables the **Remove** and **Edit...** buttons. You can also click **Add Debug Environment Variables**.



Any Variables that you add will be displayed in the Environment field of the **Submit Job** dialog box, if you have also enabled Show Advanced options.

Use Batch Extract for PBS

Selecting Show advanced options for PBS also show the Use batch extract fields.



See the discussion on Running Maxwell from a Command line for a discussion of the solve information available through batch extract.

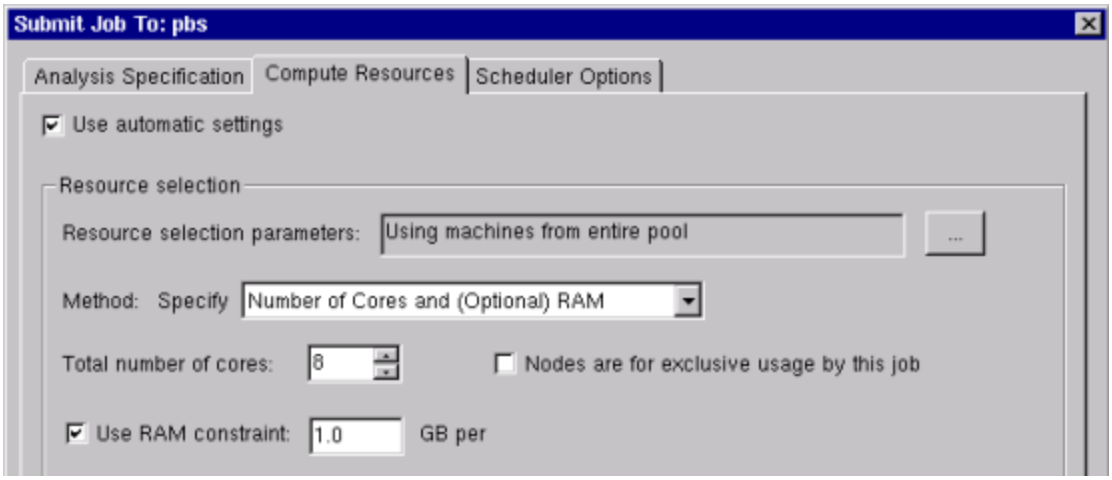
The **Preview Submission** button opens a window that shows the text commands that will be sent to the scheduler.

Compute Resource Selection for PBSPro and PBS/Torque

This section outlines how users will be able select compute resources for PBSPro and PBS/Torque jobs using the GUI. Selections include:

- Use automatic settings. If you do not select this, you specify task and/or core parameters. If you do, these task/core assignments are made automatically.
- Resource selection parameters.
- Method - drop down list with selections for Number of tasks and cores, or Number of cores and (optional) RAM, or Node List. Each method has a different set of associated parameters.
- Task and Core parameters. The parameters offered depend on Method selection.
- RAM parameters
- Job distribution.

If you select Use automatic settings with the Number of Cores and (Optional) RAM, the parameters include Total number of cores, exclusive usage option, and RAM parameters. .



Note	If you select Use automatic settings, the solver does not support automatic distribution of variations. The Use automatic settings option does not support Optimetrics variations. It does distribute frequencies, domains, and use of multiple level domains.

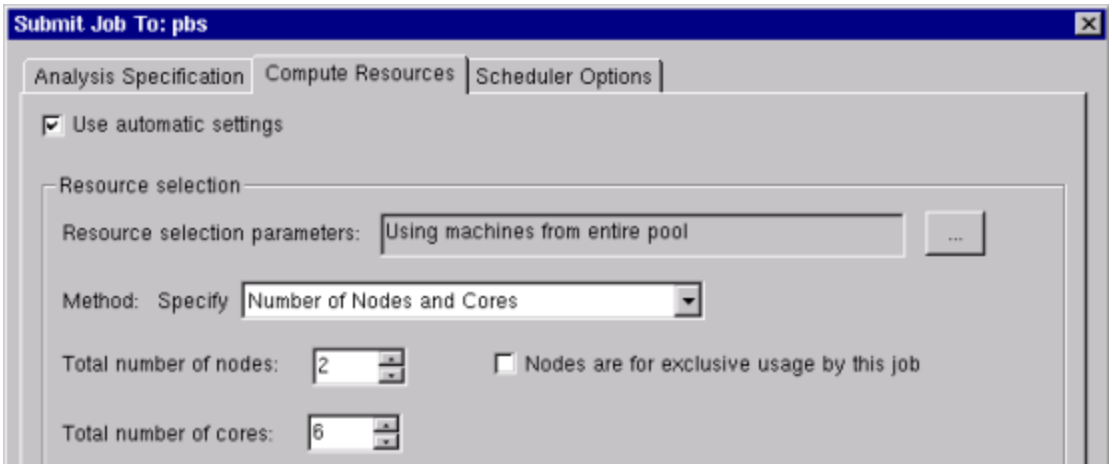
If you select Use automatic settings and Number of Cores and (Optional) RAM, and do not select Use RAM constraint, the text field is disabled

RAM Constraint

If you check **Use RAM Constraint**, you can specify a GB RAM constraint.

In response to a set of minimal constraints, the Scheduler may increase the resources assigned beyond the minimal values in order to meet the full set of requirements. Notice that a preview of the Submit Job Results shows the number of resources assigned, and that the scheduler generated code includes an MPI specification.

If you select Use automatic settings and Number of Nodes and Cores as the Method, the choices are for Number of Nodes and Cores.

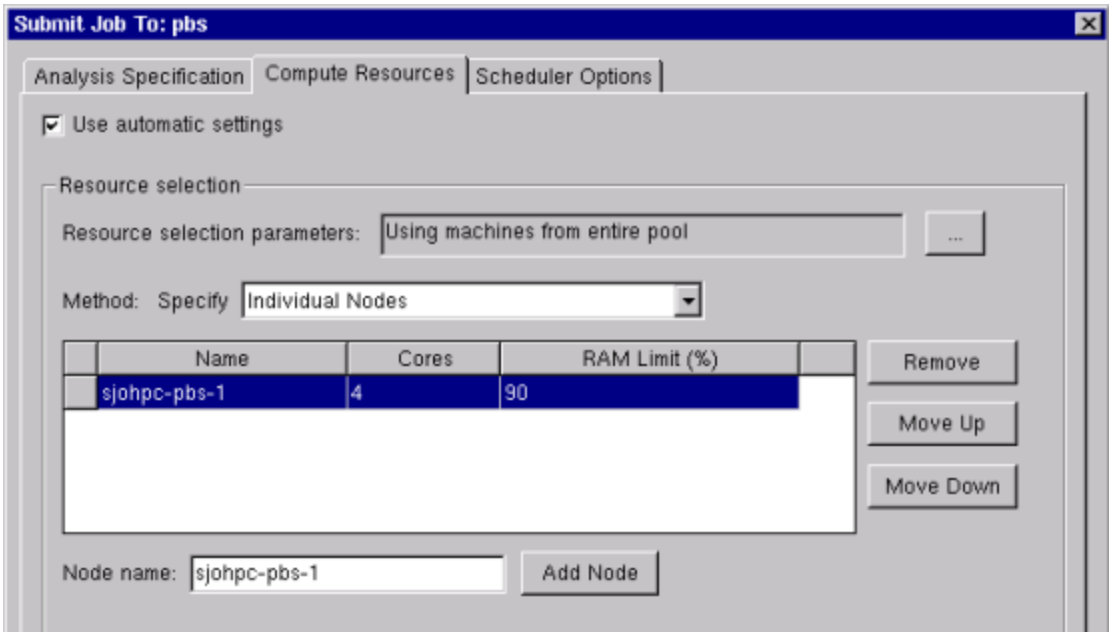


No User Specified Node List

If you do not specify the "Individual nodes" method, then the node list edit control is disabled. The user must specify resource selection parameters for nodes, cores, and tasks, depending on the method selected. If the server is the default server, then an attempt is made to determine if sufficient resources to run the job are up and available to the job on the server before it is submitted. If the server is the default server, and the queue is an execution queue, then an attempt is made to determine if sufficient resources to run the job are up and available to the job in the queue before the job is submitted. If either check fails, the job is not submitted and an error message is returned. If the server is not the default server, then there are no checks for sufficient resources before attempting to submit the job.

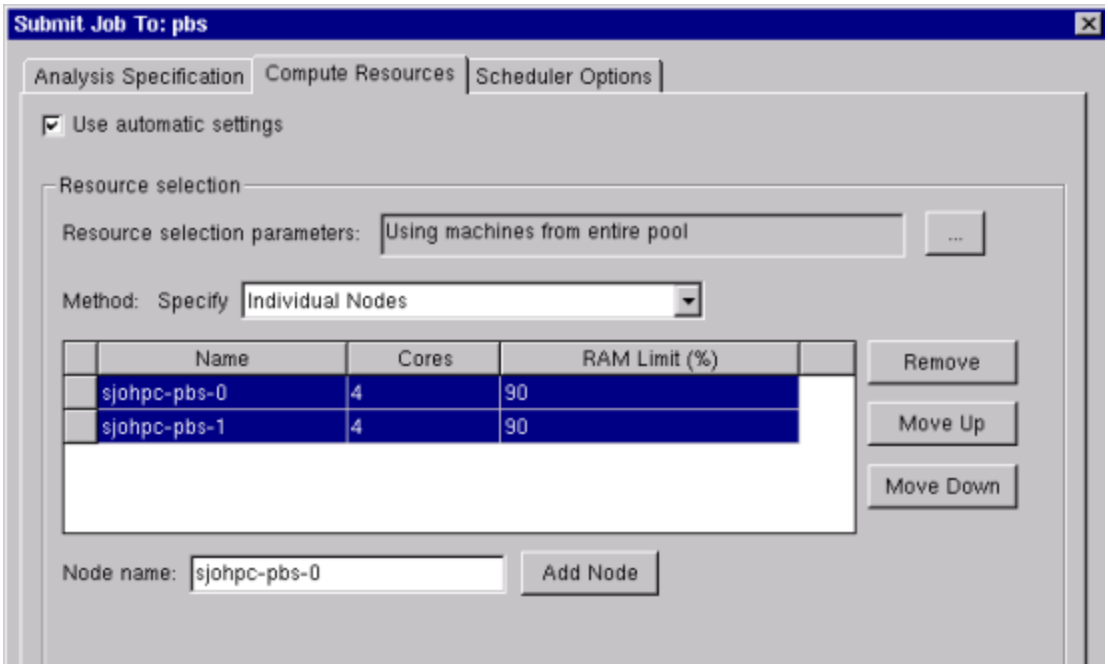
User Specified Node List for PBS

For PBSPro and PBS/Torque, a user specified explicit node list is supported when you select Individual nodes method.



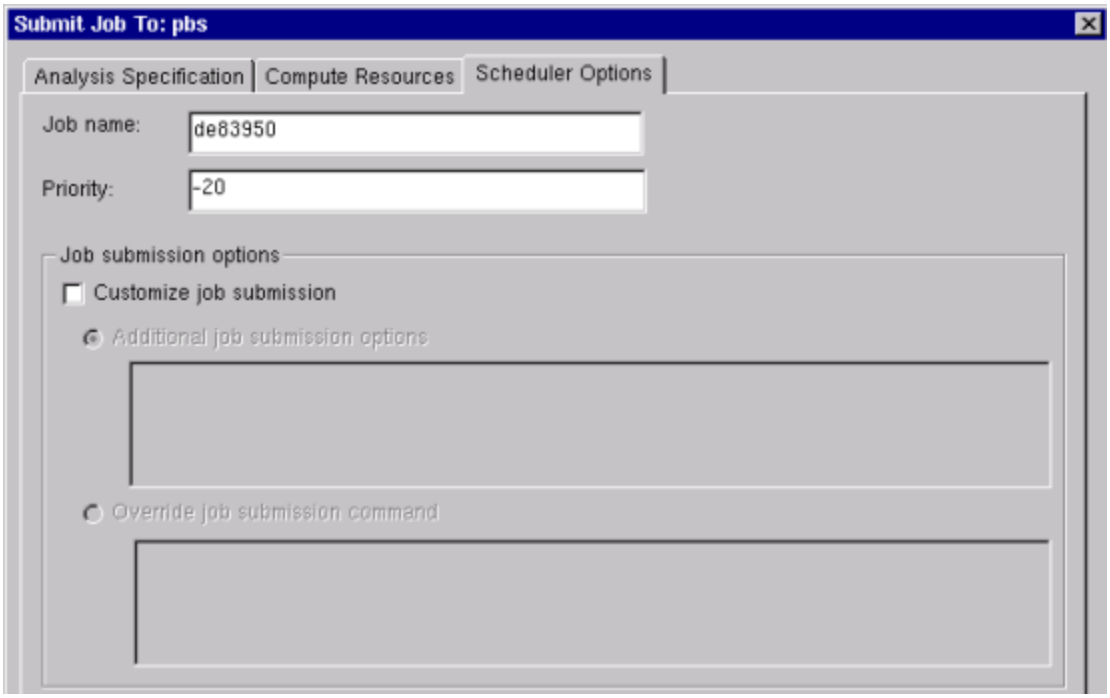
With Individual nodes selected, then you may enter a list of nodes with the number of cores and RAM limit percentage. In this case, the qsub command requests exclusive access to each of these nodes for the job. If the server for the job is the default server, then the node names are validated against the execution nodes for the default server. If any of the requested nodes are not available for new jobs, then the job is not submitted and an error message is returned. If the specified (or default) queue is an execution queue, and any of the requested nodes are not available to this queue, then the job is not submitted and an error message is returned. If the server is not the default server, then the node names are not validated by the scheduler proxy. The total number of tasks and the cores per distributed task may not be specified in this case.

If you specify multiple individual nodes, you can use the Move Up and Move Down buttons to control the order. Use the Remove button to delete the selected node.



Scheduler Options tab for PBS

On the Scheduler options tab, you can specify a job name, and set a priority.



Related Topics

Integration with PBS (Portable Batch System) Professional from Altair Engineering

Compute Resource Selection Parameters for PBS

The user may specify the following compute resource selection parameters using the grid control in the GUI:

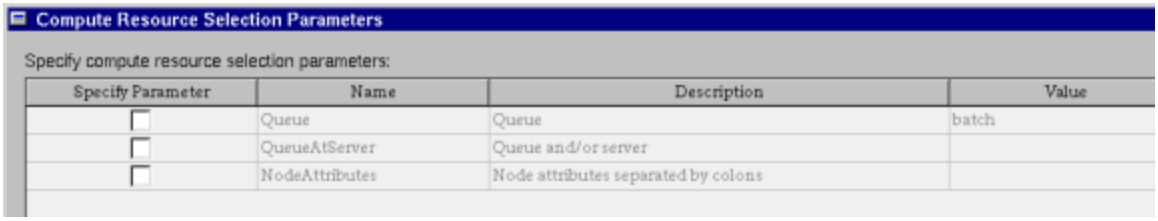
- Queue
- QueueAtServer
- NodeAttributes

These parameters are described in more detail below.

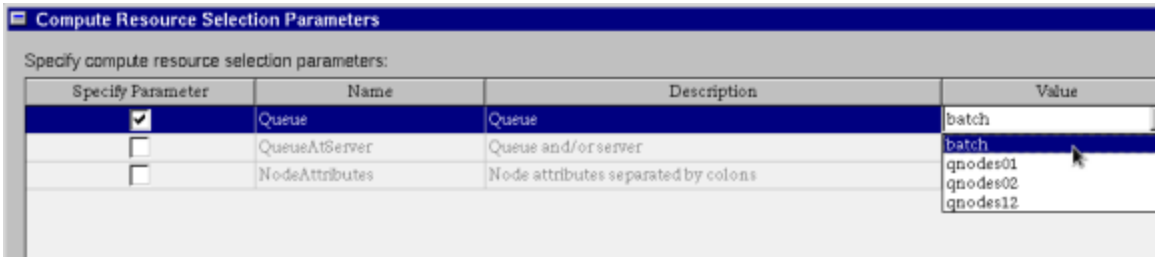
Queue

The user can select a queue for the job from the list of queues configured for the default server. Only queues that are enabled and that do not have the `from_route_only` attribute set to `true` are listed.

The following image shows the **Compute Resource Selection Parameters** dialog with no parameters specified:



The example below shows the user selecting the queue **workq** from a dropdown list containing all of the queues available at the default server.



Either the Queue or the QueueAtServer parameter may be specified, but both may not be specified. If neither the Queue parameter nor the QueueAtServer parameter is specified, then the job is submitted to the default queue at the default server.

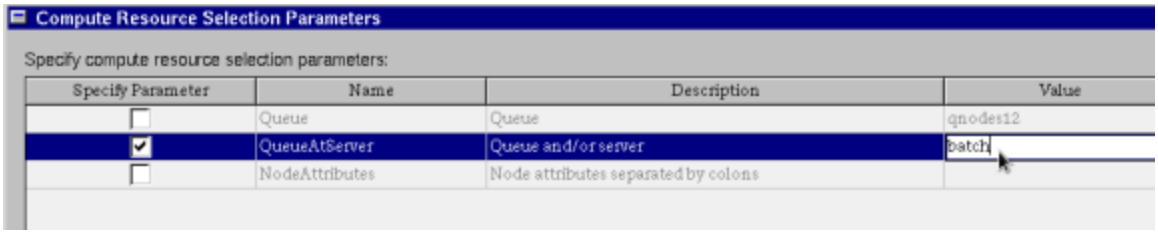
QueueAtServer

The user may specify a queue at the default server or at another server. The user may enter any text into this field. The user may enter a string in one of the following forms:

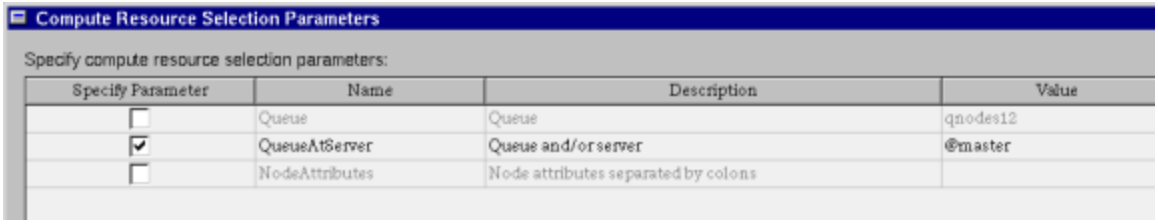
1. queue_name
2. @server_name
3. queue_name@server_name.

Using the first form, the job is submitted to the queue with name queue_name at the default server. Using the second form, the job is submitted to the default queue at the server with name server_name. Using the third form, the job is submitted to the queue with name queue_name at the server with name server_name. The *destination* value of the `-q` *destination* option on the `qsub` command line is the user specified string. This string will not be validated by the scheduler proxy library.

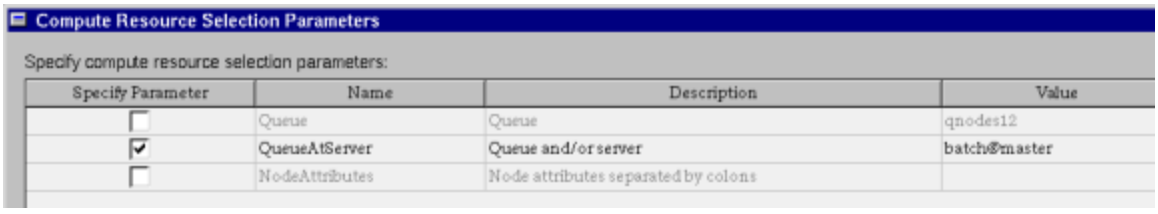
The following example shows how to enter a destination in which only a queue is specified:



The example below shows how to enter a destination in which only a server is specified:



Finally, the following example shows how to enter a destination in which both the queue and server are specified:



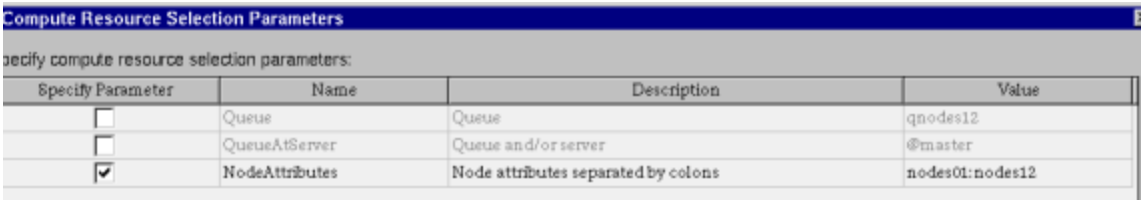
Either the **Queue** or the **QueueAtServer** parameter may be specified, but both may not be specified. If neither the **Queue** parameter nor the **QueueAtServer** parameter is specified, then the job is submitted to the default queue at the default server.

NodeAttributes

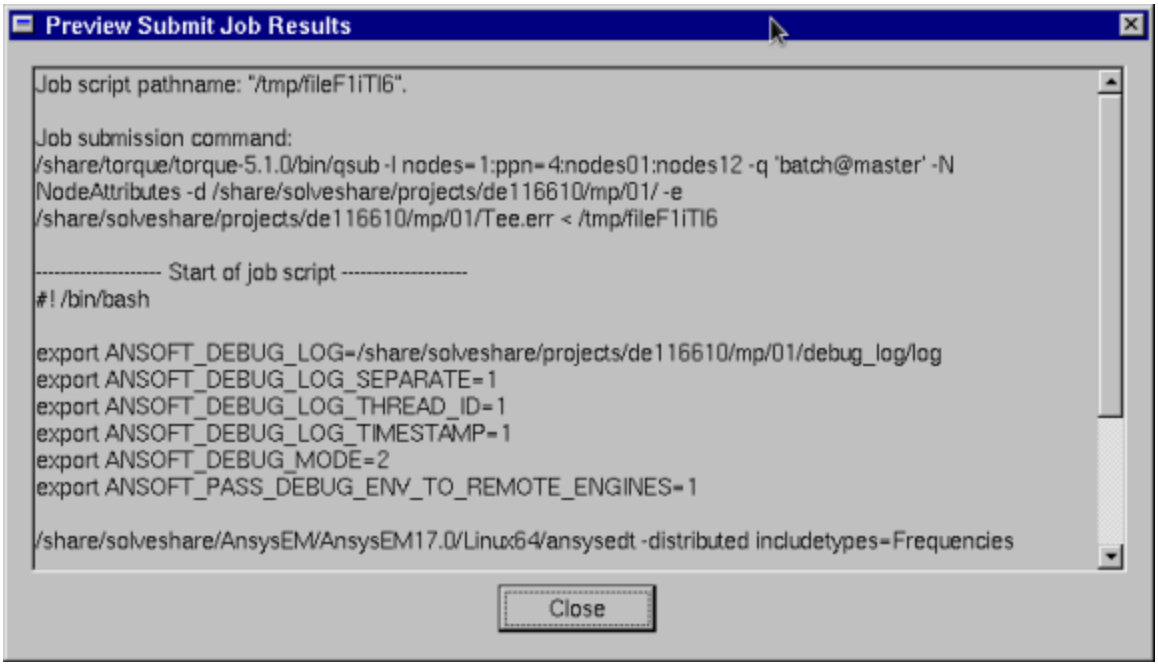
You can specify additional node attributes for the job using the **NodeAttributes** parameter. Any text can be entered into this field. If you specify multiple attribute, they should be separated by the colon character.

- For PBSPro, the contents of this field are appended to each chunk in the select resource specification of the job submission command. You can specify any resources that are valid for chunks inside a select resource specification.
- For PBS/Torque, the contents of this field are appended to each chunk in a nodes resource specification of the job submission command. In addition, only nodes that match all node features (or properties) included in the **NodeAttributes** string can be used by this job.

These are the only nodes considered when the software distributes cores to nodes. The following image shows how to specify two node attributes separated by a colon character.



The following image shows Job Preview output, which indicates how the node attributes appear in the job submission command for a PBS/Torque cluster.



Related Topics

Integration with PBS (Portable Batch System) Professional from Altair Engineering

Monitoring PBS Batch Jobs

You can monitor jobs through the Electronics Desktop user interface, or through the command line.

The suggestions below are for batch jobs run under PBS.

PBS qstat Command

The PBS **qstat** command may be used to display information on jobs and queues. In this section, several qstat command line options that may be used to monitor job progress are described.

- The **qstat -a** command displays information about all jobs in the system.
- The **qstat -r** command displays information about all running jobs in the system.
- The **qstat -s** command resembles the **qstat -r** command; the only difference is that a comment from the scheduler or batch administrator is also shown for each job.
- The **qstat -au** *userid* command displays information about all jobs owned by user userid.
- The **qstat -f** *jobid* command displays all available information about the job with id *jobid*.

See the PBS manual pages for more information.

Ansys Electromagnetics -monitor Command Line Option for PBS

The Ansys Electromagnetics **-monitor** command line option enables batch job output to the standard output and standard error streams. The warning, info, and progress messages are sent to the standard output stream. The error and fatal messages are sent to the standard error stream.

The PBS scheduler redirects the standard output and standard error streams of batch jobs to files specified in the **qsub -o** [*hostname*:]*pathname* and the **-e** [*hostname*:]*pathname*command line options, respectively. If either option is not specified, then the associated stream is redirected to the default file pathname for that stream.

The **qsub -j join** option controls whether the standard error stream for the job will be merged with the standard output stream for the job. A join value of **oe** indicates that the interleaved standard output and standard error will be sent to the standard output file or stream. A join value of **oe** indicates that the interleaved standard output and standard error will be sent to the standard error file or stream. A join value of n indicates that the standard output and standard error streams will not be joined. If the **qsub -j join** option is not specified, then the standard error and standard output streams will not be joined.

A user can monitor the progress of a job by checking the standard output file for progress, info and warning messages, and checking the standard error file for error and fatal messages.

Related Topics

Integration with PBS (Portable Batch System)

qsub Arguments

Example PBS qsub Command Lines

qsub Arguments

The PBS **qsub** command has a large number of options for control of the submission process. In this section, we review the -l nodes=*value*command line option with Ansys Electromagnetics parallel batch jobs.

This option or directive has the following format:

```
-l nodes=node_spec[+node_spec...][#suffix]
```

where *node_spec* is one of the following

```
nodename[:pc_spec[:pc_spec...]]
```

Host name of the specified node, followed by optional **ppn** or **cpp** specifiers.

```
[N][:property[:property...]][:pc_spec[:pc_spec...]]
```

Optional number of nodes, followed by optional node properties, followed by optional **ppn** or **cpp** specifiers. If the number *N* is omitted, then the default value of 1 host is used.

Here, the optional **ppn** or **cpp** specifiers *pc_spec* are of form:

```
ppn=X
```

Number of processes (tasks) per node. Default is 1 if not specified.

```
cpp=Y
```

Number of CPUs (threads) per process. Default is 1 if not specified.

The optional global suffix, #suffix, which applies to all hosts has one of the following values:

```
#excl
```

This suffix requests exclusive access to the allocated nodes.

```
#shared
```

This suffix requests shared (i.e., non-exclusive) access to the allocated nodes.

The total number of requested processes is determined by adding up the product of the number of nodes and the number of processes per node for each *node_spec*. In general, this should match the number of distributed engines specified in the Ansys Electromagnetics desktop -Machinelist num=*num_distributed_engines*command line option.

The number of CPUs per process (**cpp**) specified in the PBS **qsub** command line or in the PBS directives in the script file should generally match the number of processors per engine specified in the Desktop **-batchoptions** value.

See the PBS documentation for a complete list of options for the **bsub** command, and further information on running multi-node jobs.

Related Topics

Integration with PBS (Portable Batch System) Professional from Altair Engineering

Monitoring Ansys Electromagnetics PBS Batch Jobs

Example PBS qsub Command Lines

Example PBS qsub Command Lines

All of the following examples show how to submit Linux maxwell jobs on PBS, but similar command lines and job scripts will work for all Ansys Electromagnetics products. Most of the following examples are PBS "Single-node jobs." The last example is a PBS "multi-node jobs"; this examples demonstrate how to specify the allocation of threads, tasks and nodes to a job.

Serial job:

```
qsub ~/pbs_scripts/OptimTee.sh
```

Job Script File:

```
#!/bin/sh
/opt/AnsysEM/v231/ansysedt -ng -BatchSolve
~/projects/OptimTee.aedt
```

Serial job that needs a minimum of 4GB memory and two hours of real (wallclock) time:

```
qsub ~/pbs_scripts/OptimTee.sh
```

Job Script File:

```
#!/bin/sh
#PBS -l walltime=2:00:00
#PBS -l mem=4gb
/opt/AnsysEM/v231/ansysedt -ng -BatchSolve
~/projects/OptimTee.aedt
```

Multi-processing job using 4 cores:

```
qsub ~/pbs_scripts/OptimTee.sh
```

Job Script File:

```
#!/bin/sh
#PBS -l ncpus=4
/opt/AnsysEM/v231/ansysedt -ng -BatchSolve -batchoptions
"'Maxwell3D/Preferences/NumberOfProcessors'=4"
~/projects/OptimTee.aedt
```

- The #PBS -l ncpus=4 directive indicates that four cores or CPUs are allocated to this job.
- The -batchoptions option indicates that Maxwell should use four cores for multi-processing.

Distributed processing job using 4 engines on a single host:

```
qsub ~/pbs_scripts/OptimTee.sh
```

Job Script File:

```
#!/bin/sh
#PBS -l ncpus=4
/opt/AnsysEM/v231/ansysedt -ng -BatchSolve -Distributed
-machinelist num=4
~/projects/OptimTee.aedt
```

- The #PBS -l ncpus=4 directive indicates that four cores or CPUs are allocated to this job.
- The -Distributed option indicates that this is a DSO job, so that multiple engines will be started. Because 4 cores are allocated to the job, the job will run 4 engines. The -Distributed option may now have additional options, such as includetypes=xxx, excludetypes=xxx, maxlevels=n, and numlevel1=n, where n indicates and integer, and xxx indicates a list of distribution types or "default".

Distributed processing and multi-processing job using 8 cores on two nodes, running 4 engines (two per node) with 2 cores for multi-processing:

```
qsub ~/pbs_scripts/OptimTee.sh
```

Job Script File:

```
#!/bin/sh
#PBS -l nodes=2:ppn=2:cpp=2#excl
/opt/AnsysEM/v231/ansysedt -ng -BatchSolve -Distributed
-machinelist num=4 -batchoptions
~/projects/OptimTee.aedt
```

- The PBS directive **#PBS -l nodes=2:ppn=1:cpp=2#shared** indicates that two nodes are requested [2], two processes (engines) run on each node [ppn=2], and each process will use two cores [cpp=2]. The hosts allocated to this job may not be used for any other jobs while this job is running [#excl].
- The **-machinelist num=4** option indicates that this is a DSO job and that a total of four engines will be started.

Related Topics

Integration with PBS (Portable Batch System) Professional from Altair Engineering

qsub Arguments

Monitoring Ansys Electromagnetics PBS Batch Jobs

Example PBS qsub Command Lines

Integration with SLURM Scheduler (Linux Utility)

For Linux customers, this release supports the SLURM (Simple Linux Utility for Resource Management) scheduler for job submission, monitoring, and control (abort). In addition, this provides Windows-to-Linux support for the same functionality, that is submit, monitor, and control. Using SLURM as a starting point, customers should be able to develop their own integrations for other schedulers. Ansys Electronics Desktop GUI can also check job states in the SLURM scheduler within current job monitoring. Persistence mode provides more efficient management between Desktop and scheduler proxies. Windows to Linux submission also provides convenient access to generic scheduler in Linux environments through Remote RSM Service.

When you submit a job to the SLURM scheduler or Remote RSM to a SLURM cluster, you can use their own job options to override job options in the **Scheduler Options** tab. If you want to do that, the override job options must have "--export=None".

Consult the *Linux Installation Guide* for additional information on supported schedulers.

See Integrating Ansys EM Tools with Third Party Schedulers for more information.

Configuration of the generic scheduler with SLURM

Requirements

Files must be under ${ANSYS_EM_INSTALL_DIR}/schedulers directory. Desktop and RSM service use the same required file list below.

The required files are listed here:

- ${NSTALL_DIR}/schedulers/libgeneric_scheduler.so
- ${NSTALL_DIR}/schedulers/proxies.cfg
- ${NSTALL_DIR}/schedulers/scripts/persistence.py
- ${NSTALL_DIR}/schedulers/scripts/SLURM_GetInfo_linux.py
- ${NSTALL_DIR}/schedulers/scripts/SLURM_GetRunInfo_linux.py
- ${NSTALL_DIR}/schedulers/scripts/SLURM_SubmitJob_linux.py

- ${NSTALL_DIR}/schedulers/scripts/SLURM_ControlJob_linux.py
- ${NSTALL_DIR}/schedulers/scripts/SLURM_GetJobState_linux.py
- ${NSTALL_DIR}/schedulers/scripts/SLURM_LaunchProcess_linux.py
- ${NSTALL_DIR}/schedulers/scripts/utils/ComputeResourceList.py
- ${NSTALL_DIR}/schedulers/scripts/utils/ErrorCode.py
- ${NSTALL_DIR}/schedulers/scripts/utils/__init__.py
- ${NSTALL_DIR}/schedulers/scripts/utils/RunCommand.py
- ${NSTALL_DIR}/schedulers/scripts/utils/slurm-intelMPI-integration.sh

Environment Variables

There are some environment variables for the Generic Scheduler proxy. All environment variables apply on both Desktop and RSM Service.

ANSYSEM_HPC_MPI_USE_SSH

MPI tight integration is disabled if this variable is set. If this environment variable not set, the proxy will use tight integration to run jobs. If MPI tight integration needs to be disabled, set this environment variable before launching Ansys Electronics Desktop to submit jobs, and submit this environment variable to job execution nodes with job configuration.

ANSYSEM_GENERIC_SCRIPT_DIR

This variable adds a customized directory to be searched for proxy scripts. The path can be any form of following addresses. Persistence script path will not change.

- absolute address to any folder with read access
- relative address to folder under the Electronics Desktop installation directory
- relative address to folder under the schedulers directory
- relative address to folder under the schedulers/scripts directory

ANSYSEM_GENERIC_EXEC_PATH

This variable adds customized directories to be searched for proxy executables. Mono executable and IronPython executable should be specified when they are not in the Ansys Electronics Desktop installation. This environment variable must be set for RSM Service.

ANSYSEM_GENERIC_PROXY_PERSISTENCE

If set, this variable redirects the persistence script location to be used; otherwise, the default persistence script path is used.

Configuration File

The configuration file "proxies.cfg" is located at ${ANSYS_EM_INSTALL_DIR}/schedulers/. The file can have multiple scheduler configurations. The file content and explanation are below.

[slurm] ***# Unique scheduler name***

DisplayName=SLURM ***# Display name in GUI***

 # Proxy Scripts

SubmitJob=scripts/SLURM_SubmitJob_linux.py

ControlJob=scripts/SLURM_ControlJob_linux.py
GetSchedInfo=scripts/SLURM_GetInfo_linux.py
GetJobState=scripts/SLURM_GetJobState_linux.py
GetRunInfo=scripts/SLURM_GetRunInfo_linux.py
LaunchProcess=scripts/SLURM_LaunchProcess_linux.py
 # Persistence Mode
Persistence=Yes

Persistence Mode

Persistence mode uses a persistent process to communicate with Ansys Electronics Desktop and proxy scripts. If persistence mode is disabled, Ansys Electronics Desktop uses proxy scripts directly. Enabling persistence mode may help speed up communication with proxy scripts.

> **Note:** Disabling persistence will cause large delays in GUI functionality. The primary use case of disabling persistence is to support troubleshooting.

- To enable persistence mode, make sure "Persistence=Yes" is in configuration file before launching Ansys Electronics Desktop.
- To disable persistence mode, make sure "Persistence=No" is in configuration file before launching Ansys Electronics Desktop.

Remote RSM Service Support

In ansoftrsmservice.cfg, fill your scheduler information from configuration file proxies.cfg as below. SchedulerName must be generic. Change "Proxy" as needed.

```
$begin 'Scheduler'
'SchedulerName'='generic'
'ConfigString'='{"Proxy":"slurm"}'
$end 'Scheduler'
```

Then, set the environment variable “ANSYSEM_GENERIC_EXEC_PATH” with Mono and IronPython path. Multiple paths can be filled with delimiter “:”. Ansys Electronics Desktop has built-in components in AnsysEM installation path, and you can replace the installation path like:

```
ANSYSEM_GENERIC_EXEC_PATH= <AnsysEM
Installation>/Linux64/common/mono/Linux64/bin:<AnsysEM
Installation>/Linux64/common/IronPython
```

Also set environment variable “ANSYS_EM_GENERIC_COMMON_TEMP” to a directory widely accessible to users using Remote RSM Service on the cluster. This environment variable specifies a temporary folder to use. Thus, it must be accessible for most users.

```
ANSYS_EM_GENERIC_COMMON_TEMP=/tmp
```

Then, launch **ansoftrsmservice** in the AnsysEM RSM installation folder. (Be sure not to confuse AnsysEM (Ansoft) RSM with ANSYS RSM since these are two very different things.) It is

recommended that the user for **ansoftrsmservice** should be a dedicated service user account for only running **ansoftrsmservice**. This account should have limited permissions for security reasons. The service user should not have a login. It is recommended to disable login, so that the only way to run as the user is to “su” to that user from the root (or other accounts with sudo permissions). Other users can use any account credentials in the Desktop Select Scheduler GUI to submit jobs as the requested user.

Using the generic scheduler with SLURM

To select scheduler and get scheduler information:

1. Click **Tools** > **Job Management** > **Select Scheduler...**, or select the **Simulation** tab on the ribbon, and click the **Scheduler** icon.

   This opens the **Select Scheduler** dialog box.
2. Select **SLURM** from the drop-down menu; or, if the Remote RSM service is used, fill in scheduler master node address, and use valid account credentials.
3. Click **Refresh**.

   Scheduler Information should show.
4. Click **OK** to select the scheduler.

   Submit Jobs and Monitor Jobs with SLURM

   Job submission and monitoring follow the same process and method as using other schedulers. Currently, submission using a node-list is not supported.
5. Use **Tools** > **Job Management** > **Monitor Job** to monitor the job's progress.

Using MPI Tight Integration

By default, jobs would be submitted with SSH as remote spawn command method. If MPI tight integration preferred, you should submit the job with an additional batch option setting in the Submit Job Browser:

Enable MPI Tight Integration:

```
<design-type>/RemoteSpawnCommand = “Scheduler”
```

Enable SSH:

```
<design-type>/RemoteSpawnCommand = “SSH”
```

Related Topics:

Selecting a Scheduler

Submitting a Job

Monitoring Jobs

Running Ansys Electronics Desktop from the Command Line

Command Line Information for HPC

Ansys Electromagnetics Jobs

For most cluster environments, an Ansys Electromagnetics job will consist of an Ansys Electromagnetics Desktop running in non-graphical mode, performing a batch solve. The user will submit the job to the scheduler, specifying an Ansys Electromagnetics Desktop command line to be executed on the cluster. For some schedulers, the user may or must specify a script to run instead of specifying the Ansys Electromagnetics Desktop command line; in these cases, the script will contain the corresponding Ansys Electromagnetics Desktop command line. When the resources requested for the job are available to the job, the scheduler will start the job. In many cases, the user submitting the job will not know which host or hosts are allocated to the job. With direct integration, if the Ansys Electromagnetics job is a distributed job, the Ansys Electromagnetics Desktop will query the scheduler for the hosts allocated to the job, and it will use the scheduler facilities to launch the distributed engines.

Related Topics

High Performance Computing (HPC) Integration

Running Maxwell from a Command Line

Command Line Information for Ansys Electromagnetics Suite Desktop Products

Any **Tools>Options** setting can be specified via command line, using corresponding registry keys.

This feature is available in all desktop products.

Examples

```
ansysedt.exe -batchsolve
-batchoptions -machinelist num=2
"'Maxwell/HPCLicenseType'=pool
projectname.aedt
```

This example demonstrates how to set the same options as the previous example, but here using a registry.txt file:

```
ansysedt.exe -batchsolve -batchoptions registry.txt
projectname.aedt
```

Registry.txt contains:

```
$begin 'Config'-machinelist num=2
'Maxwell/HPCLicenseType'=pool
$end 'Config'
```

Distributed Jobs

An Ansys Electromagnetics batch job which distributes the analysis over several hosts may also be called a distributed job. To submit a distributed job, the following Ansys Electromagnetics desktop command line options should be used:

- The -Distributed option should be present, and the -Local option should be absent. When running as a batch job under one of the schedulers with direct integration, this option is a directive to the job to 1) obtain the list of hosts allocated to the job, directly from the scheduler, and to 2) use the scheduler to launch the analysis engines on the hosts allocated to the job.
- The -Machinelist num=*num_distributed_engines*option should be included, where *num_distributed_engines*is the total number of analysis engines to be started on the hosts assigned to the job.

Other examples:

- Serial Job on a Single Processor
- Distributed Job using Four Processors
- Multiprocessing Job Using Four Cores
- Distributed Analysis and Multi-Processing in the Same Job

Serial Job on a Single Processor

Suppose Maxwell is installed at “C:\Program Files\AnsysEM\v231\Win64\” and you are using AnsoftRSMService for remote-analysis/DSO:

```
C:\Program Files\AnsysEM\v231\win64\ansysedt.exe -ng
-BatchSolve -machinelist num=2
-monitor \\shared_drive\projs\OptimTee.aedt
```

User is using LSF for remote-analysis/DSO

```
bsub -n 1 C:\Program Files\AnsysEM\v231\win64\ansysedt.exe -ng
-BatchSolve -machinelist num=3 -monitor -local \\shared_
drive\projs\OptimTee.aedt
```

Distributed Job using Four Processors

RSM Service

```
C:\Program Files\AnsysEM\v231\win64\ansysedt.exe-ng -Batchsolve
-monitor -Distributed
-machinelist list="10.1.1.221, 10.1.1.222, 10.1.1.223,
10.1.1.224" \\shared_drive\projs\OptimTee.aedt
```

LSF

```
bsub -n 4 C:\Program Files\AnsysEM\v231\win64\ansysedt.exe -ng
-Batchsolve -monitor
-Distributed -machinelist num=4
\\shared_drive\projs\OptimTee.aedt
```

Multiprocessing Job Using Four Cores

Multi-processing job using 4 cores

```
bsub -n 4 -R "span[ptile=4]" C:\Program
Files\AnsysEM\v231\win64\ansysedt.exe -ng -monitor
-Local -BatchSolve -machinelist num=4 -batchoptions \\shared_
drive\registry.txt \\shared_drive\projs\OptimTee.aedt
```

This requests 4 cores to come from the same machine, as multi-processing needs cores to be on the same machine

Distributed Analysis and Multi-Processing in the Same Job

Distributed-processing using 4 engines and multi-processing using 4 cores, using a total of 16 cores

```
bsub -n 16 -R "span[ptile=4]" C:\Program
Files\AnsysEM\v231\win64\ansysedt.exe -ng
-BatchSolve -Distributed -machinelist num=4
-batchoptions \\shared_drive\registry.txt
\\shared_drive\projs\OptimTee.aedt
```

Related Topics

Running Maxwell from a Command Line
Integration with Platform's Load Sharing Facility (LSF)
General Terminology for LSF
What a Scheduler Does
Installation of Ansys Electromagnetics Tools on LSF Cluster
Integration of Ansys Electromagnetics Products with LSF
Known Issues for LSF
Troubleshooting for LSF
Aborting an Analysis

Integrating Ansys Electromagnetics Tools with Third Party Schedulers

This section indicates how to create a dynamically linked library to allow integration of Ansys Electromagnetics tools with an arbitrary scheduler environment. Each scheduler proxy library is used for a single specific scheduler environment. If the library is installed with a valid name and in the correct location, then it will automatically be loaded and used by Ansys Electromagnetics tools.

- Introduction
- Common Requirements for Running Jobs
- Using a Shared Library (Linux) or a DLL (Microsoft Windows)

- Scheduler Proxy Interfaces
- Using an IronPython Program for Scheduler Integration

Introduction

Ansys Electromagnetics Software Tools may be run as serial or parallel jobs on a cluster under control of a scheduler. Serial jobs are run using a single analysis engine at any one time on a single host. If the tool performs multiple analyses (for a frequency sweep or a parametric analysis, for example), the analyses are performed one after the other. Parallel jobs are run using multiple analysis engines running in parallel on the same host or on separate hosts. For parts of the analysis (such as meshing), the parallel job may use only a single analysis engine on a single host. Other parts of the analysis (such as a frequency sweep, parametric analysis or DDM, for example) may be distributed to multiple analysis engines running in parallel.

- Serial Jobs
- Parallel Jobs

Serial Jobs

When an Ansys Electromagnetics batch analysis runs as a serial job, the analysis engines run on the same host as the desktop process. The desktop process does not need to interact with the scheduler to get the names of hosts allocated to the job or to start processes on other hosts.

Parallel Jobs

For a parallel job, the desktop process starts multiple analysis engines that run in parallel. These engines may be started on the host where the desktop process is running, or on other hosts allocated to the job. The desktop process interacts with the scheduler to obtain information on the hosts that are allocated to the job, and to start engines on the local host or on other hosts allocated to the job. This document provides information on how to facilitate this interaction between the desktop process and the scheduler controlling the cluster.

For some popular job schedulers in a standard configuration, Ansys Electromagnetics provides an "out of the box" integrated solution that will work with the scheduler. In this case, the Ansys Electromagnetics installation includes code that will determine if the analysis is running as a scheduler job and communicate with the scheduler when needed. For other schedulers, the code to obtain information about the hosts allocated to a job and to distribute portions of the job to hosts assigned to the job is not provided in the installation. In order to facilitate using Ansys Electromagnetics software tools with other schedulers, the user may provide a way for Ansys Electromagnetics tools to interact with the scheduler. Currently, two general approaches are available to users.

In the first approach, the user creates a shared library (on Linux) or a dynamically linked library (on Microsoft Windows) to provide communication between the Ansys Electromagnetics tool and the scheduler. This library is loaded by the Ansys Electromagnetics tool at runtime, and if the Ansys Electromagnetics tool is running as part of a scheduler job, the Ansys Electromagnetics tool interacts with the library to get information from the scheduler, and to start additional processes on specified hosts. Each such library implements the same set of extern "C" functions needed to

mediate the interactions between the Ansys Electromagnetics tool and the scheduler. The details of these functions are described below.

In the second approach, the user creates an IronPython program to provide communication between the Ansys Electromagnetics tool and the scheduler. This program is loaded by the Ansys Electromagnetics tool at runtime, and if the Ansys Electromagnetics tool is running as part of a scheduler job, the Ansys Electromagnetics tool uses the IronPython program to get information from the scheduler, and to start additional processes on specified hosts. Each python script contains a class implementing a specified interface, which contains functions needed to mediate the interactions between the Ansys Electromagnetics tool and the scheduler. The details of the interface are described below. The IronPython interface is equivalent to the extern "C" functions used in the first approach.

Common Requirements for Running Jobs

The following requirements must be met for serial and parallel jobs to run successfully. They apply whether using "out of the box" scheduler integration or scheduler integration using a library or using an IronPython program. When we refer to host requirements, the requirements apply to all hosts that may be allocated to an Ansys Electromagnetics serial or parallel batch job.

Installation Requirements

The Ansys Electromagnetics installation directory tree must be accessible from all cluster hosts using the same path. One way to achieve this is to place the Ansys Electromagnetics installation on a shared drive that is accessible to the cluster hosts using the same pathname. On Windows, this may require the use of UNC names to refer to the installation directory. Another option is to install the Ansys Electromagnetics tool locally on each cluster host using the same local directory pathname.

Project File and Directory Requirements

The directory containing the project file must also be available from all cluster hosts using the same path. The project file and the containing directory must be readable and writable by the user account used to run the job. The controlling process for a distributed job is called the Desktop process, and it reads from and writes to the project file and other files in the same directory and its subdirectories. Although only the Desktop process reads from and writes to this directory, the Desktop process may be started on any of the hosts allocated to the job, so all hosts should have access to this directory using the same pathname.

Using a Shared Library (Linux) or a DLL (Microsoft Windows)

This section describes how to create a dynamically linked library to allow integration of Ansoft tools with an arbitrary scheduler environment. Each scheduler proxy library is used for a single specific scheduler environment. If the library is installed with a valid name and in the correct location, then it will automatically be loaded and used by Ansys Electromagnetics tools.

Installation Details

The scheduler proxy library must be installed in the schedulers subdirectory of the Ansys Electromagnetics installation directory. For example, if the Ansys Electromagnetics installation directory is C:\Program Files\AnsysEM\v231\Win64, then the scheduler proxy library must be installed in directory C:\Program Files\AnsysEM\v231\Win64\schedulers.

The scheduler proxy library base name must match "libprefix_scheduler" on Windows and "liblibprefix_scheduler" on Linux. The extension must be a valid extension for a dynamically loaded library on the platform where it is used. The scheduler proxy library name prefix libprefix shall be unique, so it does not conflict with other scheduler proxy libraries in the same directory. To avoid confusion, the scheduler proxy library name should be all lower case on OSs where file names are case sensitive.

Related Topics

Build Information for Scheduler Proxy Library

Implementation Details for Custom Scheduler Integration

Testing Your Scheduler Integration

Troubleshooting Custom Scheduler Integration

Build Information for Scheduler Proxy Library

This section contains the recommended compiler and linker settings for building a scheduler proxy library.

- 32 Bit Microsoft Windows
- 64 Bit Microsoft Windows
- Linux

32 Bit Microsoft Windows

The proxy library should be compiled and linked as a 32 bit DLL, using the following recommended compiler and linker options:

Compiler Options

- Use of MFC: Use Standard Windows Libraries
- Character Set: Use Multi-Byte Character Set [/D "_MBCS"]
- Runtime Library: Multi-threaded DLL [/MD]
- Calling Convention: __cdecl [/Gd (default)]

Linker Options:

- Create a DLL [/DLL]
- 32 bit code [MACHINE:X86]

64 Bit Microsoft Windows

The proxy library should be compiled and linked as a 64 bit DLL, using the following recommended compiler and linker options:

Compiler Options

- Use of MFC: Use Standard Windows Libraries
- Character Set: Use Multi-Byte Character Set [/D "_MBCS"]
- Runtime Library: Multi-threaded DLL [/MD]
- Calling Convention: __cdecl [/Gd (default)]

Linker Options:

- Create a DLL [/DLL]
- 32 bit code [MACHINE:X64]

Linux

The proxy library should be compiled and linked as shared library (*.so) file. The following compiler and linker options are recommended when building using gcc/g++:

Compiler Options

- Generate 32 bit code: [-m32]
- Generate position independent code, suitable for use in a shared library: [-fpic]
- Generate code compatible with pthreads library: [-pthread]

Linker Options:

- Create a shared object file: [-shared]
- Generate 32 bit code: [-m32]
- Generate position independent code, suitable for use in a shared library: [-fpic]
- Generate code compatible with pthreads library: [-pthread]

Implementation Details for Custom Scheduler Integration

Function Name Prefix

Each exported function will have a scheduler specific function name prefix. The function name prefix will be the same as the library name prefix, except that it is converted to upper case. For example, if the library name prefix is "lsf", then the function name prefix is "LSF". In the examples below, we use FN_PREFIX to denote the function name prefix.

The scheduler proxy library must provide implementations of the following extern "C" functions:

- IsProductLaunchedInYourEnvironment
- GetTempDirectory
- GetMachineListAvailableForDistribution
- LaunchProcess
- GetUseRsmForEngineLaunch

- GetThisJobID
- GetSchedulerDisplayName

IsProductLaunchedInYourEnvironment

Purpose

Determine if the program is running in the context of the scheduler for which this library was written.

Signature

```
extern "C" bool FN_PREFIX_IsProductLaunchedInYourEnvironment();
```

Arguments

None.

Return Value

Returns true if the current process is running as a job of the scheduler. Otherwise, false is returned.

Notes

For many schedulers, the presence of certain environment variables or their values may be checked to determine if the current process is running as a job of the scheduler.

GetTempDirectory

Purpose

Get the pathname of the temporary directory provided by the scheduler for the current job. The pathname is an empty string if the scheduler does not provide a temporary directory for the current job.

Signature

```
extern "C" bool FN_PREFIX_GetTempDirectory(char * buffer,
unsigned int* length);
```

Arguments

`buffer`: Pointer to a character buffer to contain the temporary directory path name or NULL.

`length`: Pointer to a location to contain the length of the buffer. Must be a valid pointer to an unsigned int.

Return Value

If argument buffer is NULL, then the required length of the buffer is stored in the location to which argument length points, and true is returned.

If argument buffer is not NULL, then the value to which argument length points (the buffer length) is checked. If it is large enough to contain the pathname of the temporary directory, including the terminal null byte, then the pathname is copied to the buffer and true is

returned. If the buffer length is insufficient for the pathname of the temporary directory, then the buffer is unchanged, and false is returned.

Notes

To get the pathname of the temporary directory, the infrastructure first calls this function with a NULL buffer, and obtains the required length of the buffer for the pathname. After creating a buffer of the appropriate size, the infrastructure calls this function again, passing the pointer to the buffer in the buffer argument and a pointer to the size of the buffer in the length argument.

GetMachineListAvailableForDistribution

Purpose

Get the list of hosts allocated to the current job. A host will appear in the list multiple times if the scheduler has allocated multiple processors or cores on the host to the job. The number of times the host appears in the list is equal to the number of processors or cores of the host that are allocated to the current job. The list is a text string containing a space separated list of hostnames.

Signature

```
extern "C" bool FN_PREFIX_
GetMachineListAvailableForDistribution(char * buffer, unsigned
int* length);
```

Arguments

`buffer`: Pointer to a character buffer to contain the list of machines available for distribution or NULL.

`length`: Pointer to a location to contain the length of the buffer. Must be a valid pointer to an unsigned int.

Return Value

If argument buffer is NULL, then the required length of the buffer is stored in the location to which argument length points, and true is returned.

If argument buffer is not NULL, then the value to which argument length points (the buffer length) is checked. If it is large enough to contain the lists of hosts, including the terminal null byte, then the list is copied to the buffer and true is returned. If the buffer length is insufficient for the list of hosts, then the buffer is unchanged, and false is returned.

Notes

To get the list of hosts for distribution, the infrastructure first calls this function with a NULL buffer, and obtains the required length of the buffer for the list. After creating a buffer of the appropriate size, the infrastructure calls this function again, passing the pointer to the buffer in the buffer argument and a pointer to the size of the buffer in the length argument.

The hostnames in the list provided by this function shall be used in calls to LaunchProcess(). These host names must be in a format that is accepted by that function. See the section below on LaunchProcess.

LaunchProcess

Purpose

Launch a local or remote process to run an analysis engine. This function is called by the Ansys Electromagnetics desktop application to launch an engine process on a specified host. The hostname is one of the names in the list provided by the GetMachineListAvailableForDistribution function. See the GetMachineListAvailableForDistribution section above. If the hostname does not refer to the local host, then this function shall use the scheduler to launch the engine on the specified host. If the hostname refers to the local host, then the engine may be started as a child process, or it may be started using the scheduler.

Signature

```
extern "C" int FN_PREFIX_LaunchProcess(const char* hostName,
const char* exePathName, const char* arg1, const char* arg2);
```

Arguments

`hostName`: The name of the host where the process is to be launched.

`exePathName`: The pathname of the analysis engine executable to be started.

`arg1`: The first argument of the analysis engine command line.

`arg2`: The second argument of the analysis engine command line.

Return Value

Returns 0 on success. Returns a non-zero value if an error occurs.

Notes

The hostName argument will be one of the hostnames provided by the function `GetMachineListAvailableForDistribution()`.

If the hostName argument is the same as the current host, then the analysis engine process may be started as a child process. If the hostName argument is not the same as the current host, then the analysis engine process will be started on the remote host using the facilities available in the scheduler environment. The command line of the analysis engine process is `exePathName arg1 arg2`. The command line arguments `arg1` and `arg2` may contain newlines, tabs, spaces or other characters that are interpreted by the command processor, such as single quote (') or double quote (") characters, or dollar signs ($). Newlines or tabs may be replaced by spaces, if the newline or tab characters cannot be easily handled. If the analysis engine command is processed by a shell, then it may be necessary to quote any special characters in the `exePathName` or in the arguments so that the special meaning is removed. If a scheduler command is used to request the scheduler to launch the command to start the engine process, the analysis engine command may be processed by the shell twice: once when the scheduler command is processed, and a second time when the analysis engine process is started. If this is the case, then the quoting of special characters needs to account for two passes through the command processor.

GetUseRsmForEngineLaunch

Purpose

This function is optional. If this feature is not needed, then the function need not be implemented. Most schedulers should not need this feature.

For some schedulers, it may be desirable for the Ansys Electromagnetics RSM service to launch the engine processes instead of using the scheduler proxy library. For example, if the scheduler proxy library is limited to launching one process per host, then the scheduler proxy library may be used to launch one Ansys Electromagnetics RSM service executable per host, and the Ansys Electromagnetics RSM executable will launch all of the engine processes.

If the Ansys Electromagnetics RSM service should be used to launch engine processes for this scheduler, then this function shall be implemented and it shall return true.

If the Ansys Electromagnetics RSM service should not be used to launch engine processes for this scheduler, then this function is not required. If it is implemented, it should return false. If it is not implemented, it will be treated the same as if it was implemented and returns false.

Signature

```
extern "C" bool FN_PREFIX_GetUseRsmForEngineLaunch(void)
```

Arguments

None.

Return Value

Returns true if the Ansys Electromagnetics RSM service should be used to launch engine processes for this scheduler. Returns false if the Ansys Electromagnetics RSM service should not be used to launch engine processes for this scheduler.

Notes

This function is optional. If not implemented, then it is treated the same as if it was implemented and returns false.

GetThisJobID

Purpose

Get a string identifying the job currently running in the scheduler environment. This string is displayed to the end user to identify the job.

Signature

```
extern "C" bool FN_PREFIX_GetThisJobID(char * buffer, unsigned
int* length);
```

Arguments

`buffer`: Pointer to a character buffer to contain the Job ID or NULL.

`length`: Pointer to a location to contain the length of the buffer. Must be a valid pointer to an unsigned int.

Return Value

If argument buffer is NULL, then the required length of the buffer is stored in the location to which argument length points, and true is returned.

If argument buffer is not NULL, then the value to which argument length points (the buffer length) is checked. If it is large enough to contain the string identifying the current job, including the terminal null byte, then the job ID is copied to the buffer and true is returned. If the buffer length is insufficient for the job ID, then the buffer is unchanged, and false is returned.

Notes

To get the job ID, the infrastructure first calls this function with a NULL buffer, and obtains the required length of the buffer for the job ID. After creating a buffer of the appropriate size, the infrastructure calls this function again, passing the pointer to the buffer in the buffer argument and a pointer to the size of the buffer in the length argument.

For many schedulers, the job ID may be obtained from the value of an environment variable.

GetSchedulerDisplayName

Purpose

Get a string identifying the scheduler associated with the current scheduler proxy library. This string is displayed to the end user to identify the scheduler.

Signature

```
extern "C" bool FN_PREFIX_GetSchedulerDisplayName(char *
buffer, unsigned int* length);
```

Arguments

buffer: Pointer to a character buffer to contain the scheduler display name or NULL.

length: Pointer to a location to contain the length of the buffer. Must be a valid pointer to an unsigned int.

Return Value

If argument buffer is NULL, then the required length of the buffer is stored in the location to which argument length points, and true is returned.

If argument buffer is not NULL, then the value to which argument length points (the buffer length) is checked. If it is large enough to contain the scheduler display name, including the terminal null byte, then the scheduler display name is copied to the buffer and true is returned. If the buffer length is insufficient for the scheduler display name, then the buffer is unchanged, and false is returned.

Notes

To get the scheduler display name, the infrastructure first calls this function with a NULL buffer, and obtains the required length of the buffer for the scheduler display name. After creating a buffer

of the appropriate size, the infrastructure calls this function again, passing the pointer to the buffer in the buffer argument and a pointer to the size of the buffer in the length argument.

The scheduler display name is generally a fixed string.

Scheduler Proxy Interfaces

Scheduler proxy supports the following new graphical interface functions. The scheduler specific prefix of each function is not shown in this listing.

void Initialize(const std::string& config):

Initialize the proxy library for scheduler interaction. The **config** argument contains scheduler specific initialization information.

int CheckEnvironment(std::string& msg):

Check the environment in which the proxy library is running.

- Returns 0 (success) if the environment is appropriate for submitting jobs to the scheduler.
- Returns a non-zero error code if the environment is incorrect. If a non-zero error code is returned, an error message to display to the user is written to the msg argument.

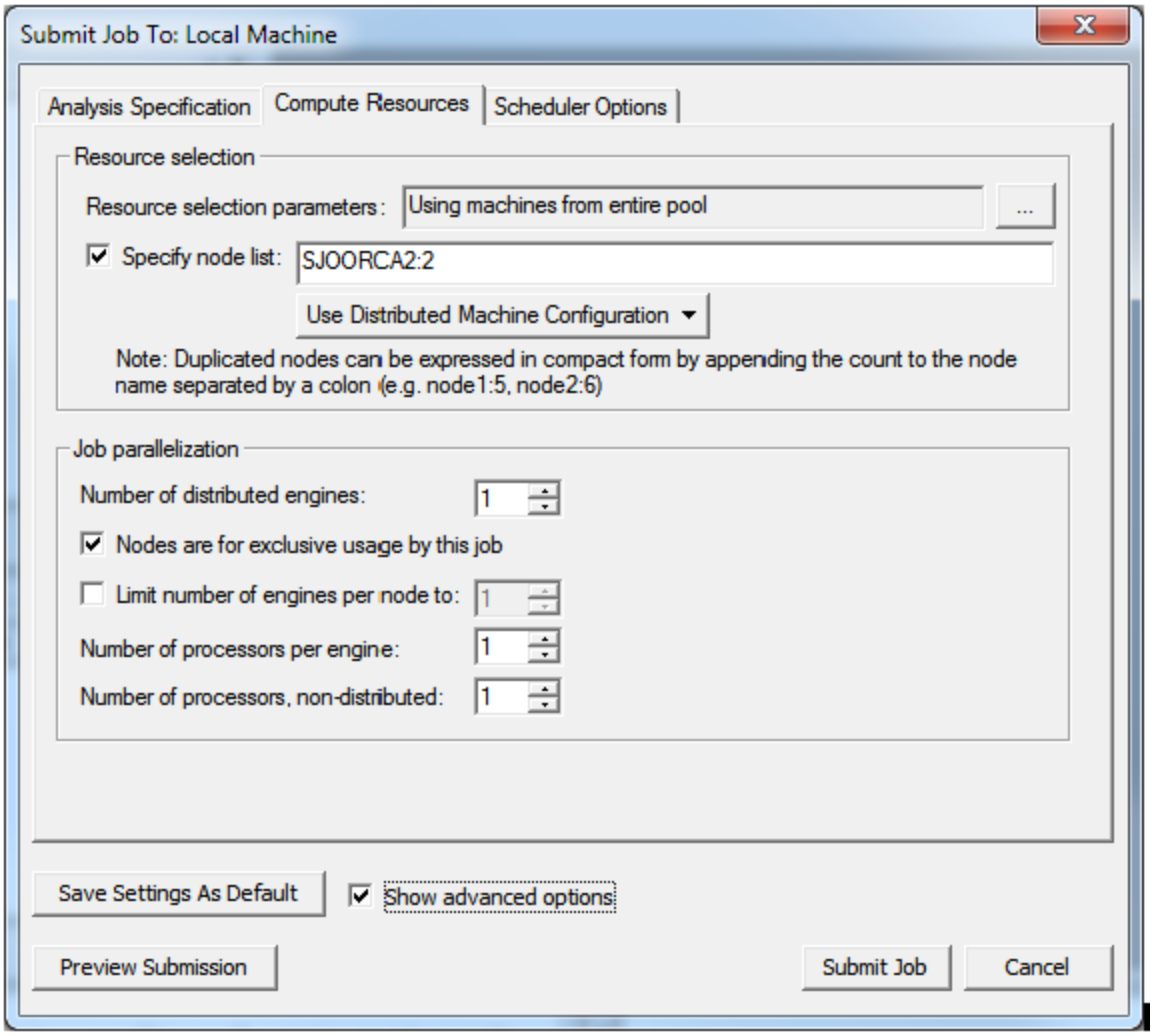
int GetSchedulerInfo(std::string& msg, std::string& schedulerName, std::string& schedulerDescription, std::string& schedulerVersion):

This function returns some basic information about the scheduler with which the scheduler proxy library interacts.

- On success, 0 is returned, and the scheduler name, scheduler description, and scheduler version are written to the **schedulerName**, **schedulerDescription** and **schedulerVersion** arguments.
- On failure, a non-zero error code is returned, and an error message to display to the user is written to the msg argument.

int GetComputeResourceAttributes(std::string& msg, AttributeDefinitionsStruct& attributeDefs):

The **Compute Resource Selection Parameters** dialog allows the user to specify scheduler specific resources. This function returns the information used to create and populate the Compute Resources tab in the Submit Job To: dialog box.



Each line in the dialog is defined by a single attribute definition in the **attributeDefs** argument. An attribute definition defines the name and description of an attribute, as well as information about the allowed values and the default value. In general, only the most commonly specified job attributes are included in the **attributeDefs** argument.

- On success, 0 is returned, and the attribute definitions are written to the **attributeDefs** argument.
- On failure, a non-zero error code is returned, and an error message to display to the user is written to the msg argument.
- If the scheduler proxy library does not support any attributes using this approach, the **attributeDefs** argument will contain no attribute definitions, and 0 will be returned.

int AbortJob(std::string& msg, const std::string& jobID, bool force, const SubmissionUserStruct& submissionUser):

This function requests the scheduler to abort a job identified by the **jobID** argument. If the force argument is true, then errors should be ignored (the exact behavior is scheduler specific). The **submissionUser** argument contains information about the client user (the

user running the Desktop process). The request to abort the job should run in the context of this user. If no user is specified, then the request to abort the job runs as the user of the process or thread running the function.

- If the request is successfully submitted, then 0 is returned.
- If there is an error, then a non-zero error code is returned, and an error message to display to the user is written to the msg argument.

int SubmitUniformJob(std::string& msg, std::string& jobID, const CmdLineStruct& cmdLineInfo, const JobParallelizationStruct& jobParallelization, const UniformComputeResourcesStruct& computeResources, const JobOptionsStruct& jobOptions, const JobAtributesStruct& jobAttributes, const SubmissionUserStruct& submissionUser,const IJobParameters* jobParametersCB):

This function submits a job to the scheduler.

- On success, 0 is returned, and the job identifier of the newly submitted job is written to the **jobID** argument.
- On failure, a non-zero error code is returned, and an error message to display to the user is written to the msg argument.

This function is used to submit jobs to the scheduler in which the resources allocated to the job are uniformly distributed across the nodes allocated to the job. All other arguments are input arguments, and they are described below:

The **cmdLineInfo** argument contains the command line arguments. The first argument is the command name.

The **jobParallelization** argument contains information on how the job should be parallelized. It contains the following integral parameters:

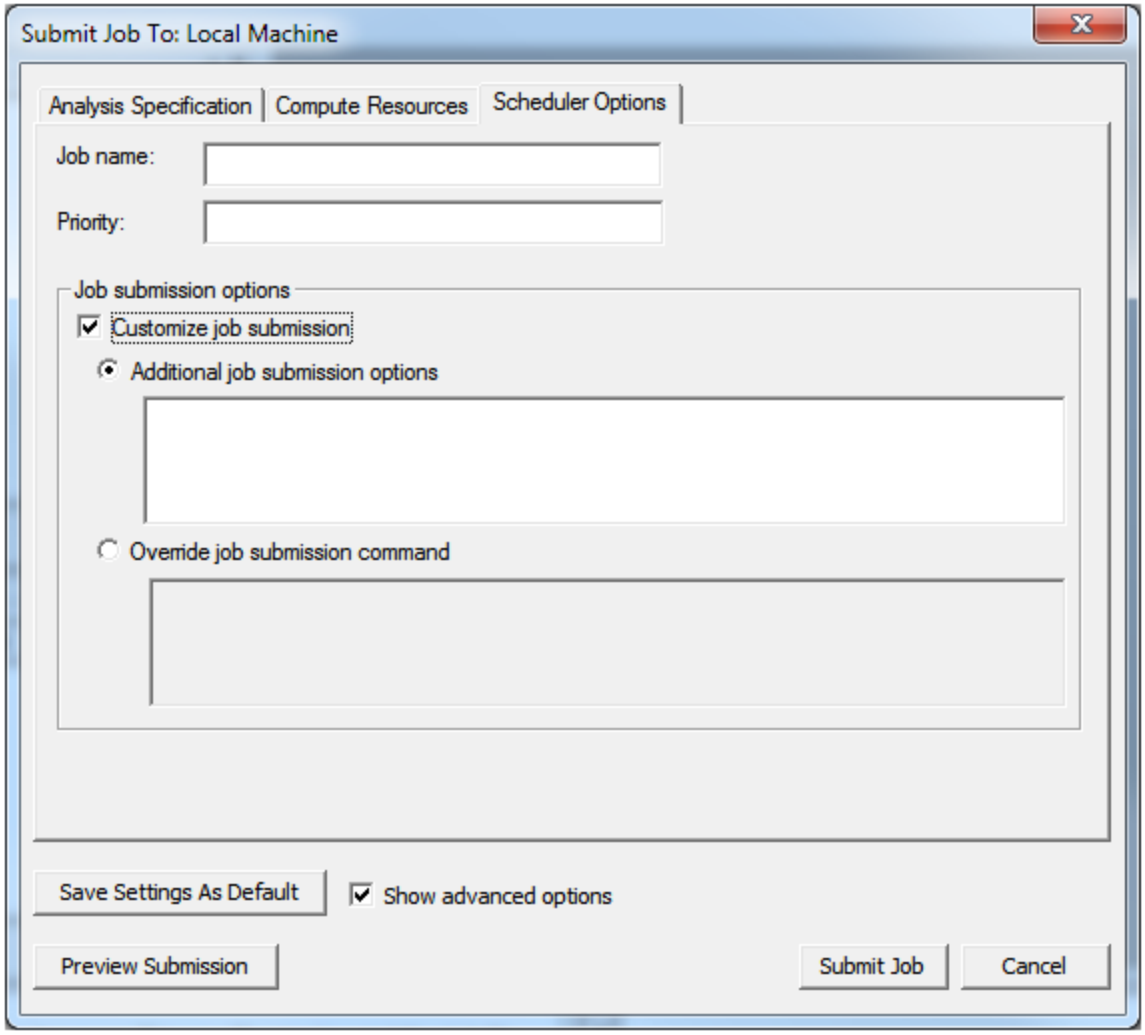
- the total number of distributed engines,
- the number of cores to allocate for each distributed engine,
- the maximum number of engines to allocate to a single node (optional), and
- the number of cores to allocated for the non-distributed portion of the analysis.
- It also contains a boolean parameter indicating whether nodes used for this job should be exclusively allocated to this job.

The **computeResources** argument is a reference to an object of type **UniformComputeResourcesStruct**. This **struct** contains zero or more resource attribute settings for the job. Each resource attribute setting consists or a resource name and a resource value. The resource name is the name of one of the resources defined in the **AttributeDefinitionsStruct** filled in by the **GetComputeResourceAttributes()** function. The resource attribute value is the value specified for the resource attribute by the user using the **Compute Resource Selection Parameters** dialog box. If no resource attributes are specified by the user in this dialog box, then the **computeResources** argument will contain no resource attribute settings.

The **jobOptions** argument contains the environment variable settings for the job.

The **jobAttributes** argument contains job submission attributes which are not necessarily related to the compute resources allocated to the job. The job name and the requested job priority are included in this data structure. The **SchedulerOptions** tab of the **Job Submission** dialog allows

the user to either specify additional job submission options or to specify all submission options, replacing the settings from the other **Job Submission** dialog controls.



The user specified submission options are included in this data structure, as well as a boolean setting indicating whether the user specified options are in addition to the automatically generated options, or whether they replace the automatically generated submission options.

The **submissionUser** argument contains information about the client user (the user running the Desktop process). The job is submitted to the scheduler to run as this user.

The **jobParametersCB** argument is a pointer to an object that implements the **IJobParameters** interface. This interface allows the scheduler proxy library to get additional information about the job. Specifically, the **GetWorkingDirectory()** interface function returns the working directory to be used for the job.

The **cmdLineInfo** argument contains the command line arguments. The first argument is the command name.

int SubmitNonUniformJob(std::string& msg, std::string& jobID, const CmdLineStruct& cmdLineInfo, const JobParallelizationStruct& jobParallelization, const NonUniformComputeResourcesStruct& computeResources, const JobOptionsStruct& jobOptions, const JobAtributesStruct& jobAttributes, const SubmissionUserStruct& submissionUser,const IJobParameters* jobParametersCB):

This function submits a job to the scheduler.

- On success, 0 is returned, and the job identifier of the newly submitted job is written to the **jobID** argument.
- On failure, a non-zero error code is returned, and an error message to display to the user is written to the msg argument.

This function is used to submit jobs to the scheduler in which the nodes to use and the number of engines to run on each node are specified by the user. All other arguments are input arguments, as for the **SubmitUniformJob()** function. These input arguments are the same as for the **SubmitUniformJob()** function, except that the **computeResources** argument is a reference to a **NonUniformComputeResourcesStruct**, as described below:

The **computeResources** argument is a reference to an object of type **NonUniformComputeResourcesStruct**. This object contains a vector of pairs, where each pair consists of the name of a node in the cluster, and the number of engines to run on the node.

int PreviewUniformJob(std::string& msg, std::string& preview, const CmdLineStruct& cmdLineInfo, const JobParallelizationStruct& jobParallelization, const UniformComputeResourcesStruct& computeResources, const JobOptionsStruct& jobOptions, const JobAtributesStruct& jobAttributes, const SubmissionUserStruct& submissionUser, const IJobParameters* jobParametersCB):

This function is similar to the **SubmitUniformJob()** function, but instead of submitting the job, text representing how the job will be submitted is written to the preview argument. Typically the preview text includes the job submission command and the contents of the job script created for the job. For some schedulers, this content may not be meaningful, so the text returned could be different.

- On success, 0 is returned, and the job preview text is written to the preview argument.
- On failure, a non-zero error code is returned, and an error message to display to the user is written to the msg argument.

The other arguments are input arguments with the same meaning as for the **SubmitUniformJob()** function. The **submissionUser** argument is ignored for this function.

Testing Your Scheduler Integration

One way to test these functions is to run the analysis for an Ansys Electromagnetics product in batch mode. When running in batch mode, a batch log file is created in the same directory as the project file. The batch log file has the same base name is the project file, with an extension of ".log". For example, if the project file is TestProject123.aedt, then the batch file is TestProject123.log. The batch log file contains useful information about the analysis run.

See the product specific help for details on running the product in batch mode, and for the command line options to use for distributed analysis.

- Testing IsProductLaunchedInYourEnvironment
- Testing GetSchedulerDisplayName and GetThisJobID
- Testing GetTempDirectory
- Testing GetMachineListAvailableForDistribution
- Testing LaunchProcess
- Testing GetUseRsmForEngineLaunch

Testing IsProductLaunchedInYourEnvironment

This function should be tested first. If the Ansys Electromagnetics application is not able to load and run this function, or if it returns false, then none of the other functions will be called. If the batch analysis is running in a scheduler environment, and this function returns true, then there will be an "info" message near the beginning of the batch log indicating that the analysis is running as a scheduler job. This message will include the scheduler display name returned by the function GetSchedulerDisplayName, and it will also include the job ID returned by the function GetThisJobID. If the batch analysis is not running in a scheduler environment, then none of the messages will include a scheduler display name or job ID.

If this message does not appear when running in a scheduler environment, ensure that the scheduler proxy library is named correctly, that it is built correctly, that it is installed in the correct directory, and that the function name prefix is the same is the library prefix converted to upper case.

Testing GetSchedulerDisplayName and GetThisJobID

As described above, when running a batch job in a scheduler environment, the scheduler display name and the job ID will appear in an "info" message near the beginning of the batch log. The values returned by these functions are copied to this message verbatim, so they can be directly compared to the expected values.

Testing GetTempDirectory

Unfinished. The temp directory displayed in the batch log is the default installation setting, not the one from the scheduler. The scheduler temp directory is set in AnsoftCOMApplication::MainFunction(), so it happens for COM engines, but not for the desktop.

Testing GetMachineListAvailableForDistribution

This function is used for distributed analysis. The analysis may be distributed across several machines if portions of the analysis are independent. For example, frequency sweeps, parametric analysis and domain decomposition allow different portions of the analysis to be

distributed across machines. The analysis in a batch job will be distributed to multiple processors or hosts if the analysis includes a setup that may be distributed (e.g., a frequency sweep or parametric analysis) and the **-Distributed** option is included in the desktop command line. The list of machines is displayed in an "info" message near the beginning of the batch log. The list in the info message can be directly compared to the expected list of machines.

To verify that the machine list is constructed correctly for a variety of cases, it may be necessary to test several jobs with different resource requirements and verify that the machine list is correct in each case. For example, one may run batch analyses with the following resource requirements:

- One processor on one host
- Several processors on one host
- One processor on each of several hosts
- Several processors on each of several hosts

Testing LaunchProcess

This function is used to launch analysis engines in the case where the analysis is distributed across multiple hosts. The analysis may be distributed across several machines if portions of the analysis are independent. For example, frequency sweeps, parametric analysis and domain decomposition allow different portions of the analysis to be distributed across machines. The analysis in a batch job will be distributed to multiple processors or hosts if the analysis includes a setup that may be distributed (e.g., a frequency sweep or parametric analysis) and the **-Distributed** option is included in the desktop command line. The list of machines is displayed in an "info" message near the beginning of the batch log. The batch log may also contain info messages when portions of the analysis distributed to different machines start or finish. These messages usually include the name of the host when the analysis ran or will run. One can verify that the analysis is actually running on the expected host or hosts using the Linux ps command or the Windows Task Manager.

In general, one analysis engine is started for each occurrence of each host in the list of machines available for distribution. For example, if the list of hosts is "hostA hostA hostA hostB hostB", then a total of 5 engines would be started, three on hostA and two on hostB. In some cases, an additional engine is started to perform the portion of the analysis which is not distributed; if this is the case, the non-distributed engine is idle during the portion of the analysis which is distributed. If this occurs in the case where the list of hosts is "hostA hostA hostA hostB hostB", then a total of 6 engines would be started, but at most 5 engines would be active at any given time. When each analysis engine is running, it may start additional child processes to do a portion of the analysis, but these are not counted as additional analysis engines because the parent of the sub-engine is inactive (waiting for the sub-engine results) when the sub-engine is active.

Testing should be sufficient to demonstrate that the scheduler proxy library can start multiple engine processes on the desktop host, and can also start multiple engine processes on other hosts.

Testing GetUseRsmForEngineLaunch

In most cases, this function will not be implemented or tested. If this function is implemented and returns true, then the Ansys Electromagnetics desktop application will not start the analysis engines using the LaunchProcess function directly. Instead, the Ansys Electromagnetics desktop application will start one AnsoftRSMService process on each host using the `LaunchProcess` function, and the engine processes will be started by these AnsoftRSMService processes. One may check for these processes using the Linux ps command or the Windows Task Manager. One AnsoftRSMService process should run on each host. These processes will be named ansoftrsmservice.exe or AnsoftRSMService.exe. These processes will be started on each host before any analysis engine is started on the host, and will remain running until the job is complete.

Troubleshooting Custom Scheduler Integration

None of the Proxy Functions are Called

There are several problems which could result in none of the proxy functions being called.

The scheduler proxy library must be installed in the schedulers subdirectory of the Ansys Electromagnetics installation directory. The installation directory is set in the registry entry `HKEY_CURRENT_USER/Software/Ansoft/Product/Version/Desktop/InstallationDirectory`, where “Product” is the Ansys Electromagnetics product name (for example, Maxwell) and “Version” is the Ansys Electromagnetics product version (for example, 14.0).

The scheduler proxy library name must match "*_scheduler.dll" on Windows and "lib*_scheduler.so" on Linux. If the library name does not match this format, then the library will not be loaded. In addition, the function name prefix must be the same as the library name prefix converted to upper case. For example, if the library name prefix is "abc", then the function name prefix is "ABC". In this example, the library name is "abc_scheduler.dll" on Windows, and "libabc_scheduler.so" on Linux. In this example, the full name of the IsProductLaunchedInYourEnvironment function is ABC_IsProductLaunchedInYourEnvironment on Windows and Linux, and it must have extern "C" linkage.

Verify that the compile and link flags follow the guidelines in the section "Build Information", above. Incorrect compile or link flags may prevent the library from being loaded by the Ansys Electromagnetics product.

If there is a problem with calling the `IsProductLaunchedInYourEnvironment` function, then none of the other functions will be called. The other functions are only called if the `IsProductLaunchedInYourEnvironment` function is successfully called and returns true.

Troubleshooting IsProductLaunchedInYourEnvironment Function

Verify that the conditions specified in the section "None of the Proxy Functions are Called" are met.

Verify that this function returns true when called in an environment running under the scheduler, and that it returns false when called in an environment not running under the scheduler.

Troubleshooting GetSchedulerDisplayName

Verify that the `IsProductLaunchedInYourEnvironment` function returns true when running in the scheduler environment.

Verify that the scheduler display name is a valid ASCII string.

Verify that, if argument buffer is NULL, then the required length of the buffer is stored in the location to which argument length points, and true is returned. The required buffer length must include space for the string null terminator.

Verify that, if argument buffer is not NULL and the value to which argument length points (the buffer length) is large enough to contain the display name, including the terminal null byte, then the display name is copied to the buffer and true is returned.

Troubleshooting GetThisJobID

Verify that the `IsProductLaunchedInYourEnvironment` function returns true when running in the scheduler environment.

Verify that the job ID is a valid ASCII string.

Verify that, if argument buffer is NULL, then the required length of the buffer is stored in the location to which argument length points, and true is returned. The required buffer length must include space for the string null terminator.

Verify that, if argument buffer is not NULL and the value to which argument length points (the buffer length) is large enough to contain the job ID, including the terminal null byte, then the job ID is copied to the buffer and true is returned.

Troubleshooting GetTempDirectory

Verify that the `IsProductLaunchedInYourEnvironment` function returns true when running in the scheduler environment.

Verify that the temporary directory name is a valid ASCII string.

Verify that, if argument buffer is NULL, then the required length of the buffer is stored in the location to which argument length points, and true is returned. The required buffer length must include space for the string null terminator.

Verify that, if argument buffer is not NULL and the value to which argument length points (the buffer length) is large enough to contain the temporary directory pathname, including the terminal null byte, then the temporary directory pathname is copied to the buffer and true is returned.

Troubleshooting GetMachineListAvailableForDistribution

Verify that the `IsProductLaunchedInYourEnvironment` function returns true when running in the scheduler environment.

Verify that the list of hosts is a valid ASCII string containing a space separated list of host names. A host name will appear in the list a number of times equal to the number of processors or cores available to the job on that host.

Verify that, if argument buffer is NULL, then the required length of the buffer is stored in the location to which argument length points, and true is returned. The required buffer length must include space for the string null terminator.

Verify that, if argument buffer is not NULL and the value to which argument length points (the buffer length) is large enough to contain the list of hosts, including the terminal null byte, then the list of hosts is copied to the buffer and true is returned.

Troubleshooting LaunchProcess

Verify that the `IsProductLaunchedInYourEnvironment` function returns true when running in the scheduler environment.

The hostName argument is a host name from the list returned by the GetMachineListAvailableForDistribution function. Verify that the LaunchProcess function can accept host names in the format returned by the GetMachineListAvailableForDistribution function.

The exePathName argument is the pathname of the analysis engine executable to be started. This pathname may contain spaces or other characters special to the shell. Ensure that the LaunchProcess function is able to handle such cases.

The arg1 and arg2 arguments may contain newlines, tabs, single quotes, spaces, dollar signs, and other characters which may be special to the shell. Ensure that the LaunchProcess function is able to handle such cases. If needed, the newline characters may be replaced by other whitespace characters. One or both of these arguments could also be an empty string; verify that the empty string is correctly passed to the engine process command line.

If a scheduler command is used to launch the engine process on a remote machine, the engine command line may be processed by the shell twice, once when the scheduler command is processed by the shell, and again when the engine command is processed by the shell. In such cases, the quoting of characters special to the shell will need to be take these two passes through the shell into account. In some implementations, it may be necessary or convenient to use different approaches for launching engine processes on the local machine and on remote machines; if this is done, verify that the approach used to determine whether the hostName argument represents the local machine is correct.

Troubleshooting GetUseRsmForEngineLaunch

In most cases, this function will not be implemented. If it is implemented, then follow the suggestions below.

`Verify that the IsProductLaunchedInYourEnvironment` function returns true when running in the scheduler environment.

If the RSM should be used for launching engines, verify that this function returns true. Otherwise, verify that this function returns false.

Using an IronPython Program for Integration with a Scheduler

This section describes how to create an IronPython program for integration with a scheduler. Each such program is used for a single specific scheduler environment. If the program is installed with a valid name and in the correct location, then it will automatically be loaded and used by Ansys Electromagnetics tools.

Installation Details

The IronPython program must be installed in the schedulers subdirectory of the Ansys Electromagnetics installation directory. For example, if the installation directory is C:\Program Files\AnsysEM\v231\Win64\, then the IronPython program must be installed in directory C:\Program Files\AnsysEM\v231\Win64\schedulers.

The program file extension must be ".py". Select the program name so that it does not conflict with other IronPython programs in the same directory. If the Operating System or file system treat file names in a case sensitive manner, the file extension ".py" must be lower case.

Python Programming Notes

The scheduler program will be run in the IronPython environment both on Microsoft Windows and on Linux. There are some differences between IronPython and CPython. The version of IronPython in use as of June 9, 2011 is 2.6.1.

Implementation Details

The program must contain the following:

Import the ISchedulerPluginExtension interface as follows:

```
from Ansys.Ansoft.SchedulerPluginDotNet import
ISchedulerPluginExtension
```

Define a class which implements the ISchedulerPluginExtension interface. In this document, this class is named SamplePluginExtension, but any class name may be used. The class member functions are described in the next section. The class definition will look similar to the following:

```
class SamplePluginExtension(ISchedulerPluginExtension):

def GetName(self):
return "SamplePluginExtension"

def GetDescription(self):
return "Example python script plugin extension"
. . .
```

Include the following line in the program so that the class that you have defined, SamplePluginExtension, is loaded by the infrastructure:

```
ExtensionRegistrar.RegisterPluginExtension
(SamplePluginExtension())
```

The infrastructure will make the EntensionRegistrar object available in the environment where the program is loaded.

Each of the functions to be implemented in the SamplePluginExtension class is described below.

- GetName
- GetDescription
- IsProductLaunchedInYourEnvironment
- GetSchedulerDisplayName
- GetThisJobID
- GetUseRsmForEngineLaunch
- GetTempDirectory
- GetMessageStringToRegisterForSigTerm
- GetMachineListAvailableForDistribution
- LaunchProcess

GetName [IronPython]

Purpose

Return a short string containing the name of the plugin extension. This string is used to identify the scheduler plugin extension in logs or program output.

Signature

```
GetName(self)
```

Arguments (excluding self)

None.

Return Value

Returns a string containing the name of the plugin extension.

Notes

The plugin extension name is generally a fixed string.

GetDescription [IronPython]

Purpose

Return a string containing the description of the plugin extension. This string is used to identify the scheduler plugin extension in logs or program output.

Signature

```
GetDescription(self)
```

Arguments (excluding self)

None.

Return Value

Returns a string containing the description of the plugin extension.

Notes

The plugin extension description is generally a fixed string.

IsProductLaunchedInYourEnvironment [IronPython]

Purpose

Determine if the program is running in the context of the scheduler for which this program was written.

Signature

```
IsProductLaunchedInYourEnvironment(self)
```

Arguments (excluding self)

None.

Return Value

Returns True if the current process is running as a job of the scheduler. Otherwise, False is returned.

Notes

For many schedulers, the presence of certain environment variables or their values may be checked to determine if the current process is running as a job of the scheduler.

GetSchedulerDisplayName [IronPython]

Purpose

Get a string identifying the scheduler associated with the current plugin extension. This string is used to identify the scheduler.

Signature

```
GetSchedulerDisplayName(self)
```

Arguments (excluding self)

None.

Return Value

Returns a string containing the description of the scheduler for which this plugin extension was written.

Notes

The scheduler display name is generally a fixed string.

GetThisJobID [IronPython]

Purpose

Get a string identifying the job currently running in the scheduler environment. This string is displayed to the end user to identify the job.

Signature

```
GetThisJobID(self)
```

Arguments (excluding self)

None.

Return Value

Returns a string containing the Job ID for the current job.

Notes

For many schedulers, the job ID may be obtained from the value of an environment variable.

GetUseRsmForEngineLaunch [IronPython]

Purpose

For some schedulers, it may be desirable for the AnsoftRSM program to launch the engine processes instead of using the scheduler plugin extension directly. For example, if the plugin extension is limited to launching one process per host, then the plugin extension may be used to launch one AnsoftRSM executable per host, and the AnsoftRSM executable will launch all of the engine processes.

If AnsoftRSM should be used to launch engine processes for this scheduler, then this function shall return True.

If AnsoftRSM should not be used to launch engine processes for this scheduler, then this function shall return False.

Signature

```
GetUseRsmForEngineLaunch(self)
```

Arguments (excluding self)

None.

Return Value

Returns True if AnsoftRSM should be used to launch engine processes for this scheduler. Returns False if the plugin extension should be used to directly launch engine processes for this scheduler.

Notes

If this function returns True, then the plugin extension will directly launch only one process on each host.

GetTempDirectory [IronPython]

Purpose

Get the pathname of the temporary directory provided by the scheduler for the current job. The pathname is an empty string if the scheduler does not provide a temporary directory for the current job.

Signature

```
GetTempDirectory(self)
```

Arguments (excluding self)

None.

Return Value

Returns a string containing the pathname of the temporary directory provided by the scheduler for the current job. Returns an empty string if no temporary directory is provided by the scheduler for the current job.

Notes

If the return value is an empty string, then the temporary directory specified on the command line or in the registry will be used.

GetMessageStringToRegisterForSigTerm [IronPython]

Purpose

Obsolete. This function should return an empty string.

Signature

```
GetMessageStringToRegisterForSigTerm(self)
```

Arguments (excluding self)

None.

Return Value

Returns an empty string.

Notes

This function should always return an empty string.

GetMachineListAvailableForDistribution [IronPython]

Purpose

Get the names of the hosts allocated to the current job. A host name will appear in the output string multiple times if the scheduler has allocated multiple processors or cores on the host to the job. The number of times the host name appears in the string is equal to the number of processors or cores of the host that are allocated to the current job. The output is a text string containing the host names separated by space characters.

Signature

```
GetMachineListAvailableForDistribution(self)
```

Arguments (excluding self)

None.

Return Value

A string containing the names of the hosts allocated to the job, separated by space characterss. The number of times the host appears in the string is equal to the number of processors or cores of the host that are allocated to the current job.

Notes

The hostnames in the string provided by this function shall be used in calls to LaunchProcess(). The host names must be in a format that is accepted by that function. See the section below on LaunchProcess.

LaunchProcess [IronPython]

Purpose

Launch a local or remote process to run an analysis engine. This function is called by the Ansys Electromagnetics desktop application to launch an engine process on a specified host. The hostname is one of the names provided by the GetMachineListAvailableForDistribution function. See the GetMachineListAvailableForDistribution section above. If the hostname does not refer to the local host, then this function shall use the scheduler to launch the engine on the specified host. If the hostname refers to the local host, then the engine may be started as a child process, or it may be started using the scheduler.

Signature

```
LaunchProcess(self, hostName, exePathName, arg1, arg2)
```

Arguments (excluding self)

hostName: The name of the host where the process is to be launched.

exePathName: The pathname of the analysis engine executable to be started.

arg1: The first argument of the analysis engine command line.

arg2: The second argument of the analysis engine command line.

Return Value

Returns 0 on success. Returns a non-zero value if an error occurs.

Notes

The hostName argument will be one of the hostnames provided by the function GetMachineListAvailableForDistribution().

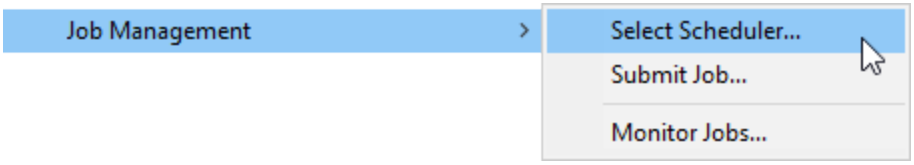
If the hostName argument is the same as the current host, then the analysis engine process may be started as a child process. If the hostName argument is not the same as the current host, then the analysis engine process will be started on the remote host using the facilities available in the scheduler environment. The command line of the analysis engine process is exePathName arg1 arg2. The command line arguments arg1 and arg2 may contain spaces or other characters that are interpreted by the command processor, such as backslash (\), single quote (') or double quote (") characters, or dollar signs ($). If the analysis engine command is processed by a shell, then it may be necessary to quote any special characters in the exePathName or in the arguments so that the special meaning is removed. If a scheduler command is used to request the scheduler to launch the command to start the engine process, and that command is processed by a command shell, then the analysis engine command may be processed by the shell twice: once when the scheduler command is processed, and a second time when the analysis engine process is started. If this is the case, then the quoting of special characters needs to account for two passes through the command processor.

The command line arguments arg1 and arg2 may be empty strings. These arguments must be preserved, even if they are empty strings. In some versions of the IronPython subprocess module, empty argument strings are discarded, resulting in an incorrect number of command line arguments. A workaround for this issue is to replace an empty string argument by a string consisting of a single space character.

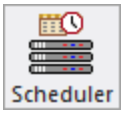
Selecting a Scheduler

Access the **Select Scheduler** window one of three ways:

- Click **Tools** > **Job Management** > **Select Scheduler...**

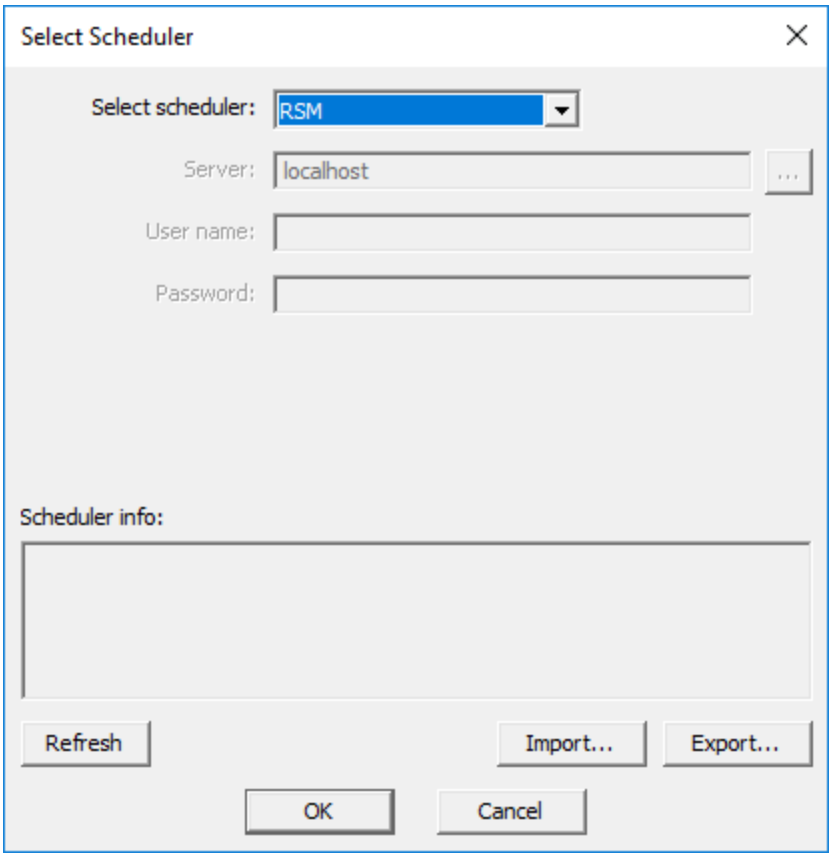


- Select the **Simulation** tab and click the **Scheduler** icon.



- From a command window, use the -showselectscheduler command.

The **Select Scheduler** window appears:



From the **Select Scheduler** window:

- Use the **Select scheduler** drop-down menu to select a scheduler.

> **Note:**
>
> If you select a scheduler that is unsupported in your environment, you will receive a warning message.
>
> See HPC Integration for a list of currently supported schedulers.

- If applicable for the scheduler type, enter server and user information.

- Information about the selected scheduler appears in the **Scheduler info** field.
- Click **OK** to complete your selection.

Submitting a Job

Ansys Electromagnetics Desktop supports its own Remote Simulation Management (RSM) and Ansys Cloud software along with other High Performance Computing (HPC) software management programs (See: HPC Integration). The **Simulation** tab of the ribbon includes icons for setting HPC Options, creating and selecting analysis configuration, selecting the scheduler, submitting jobs, and monitoring jobs.

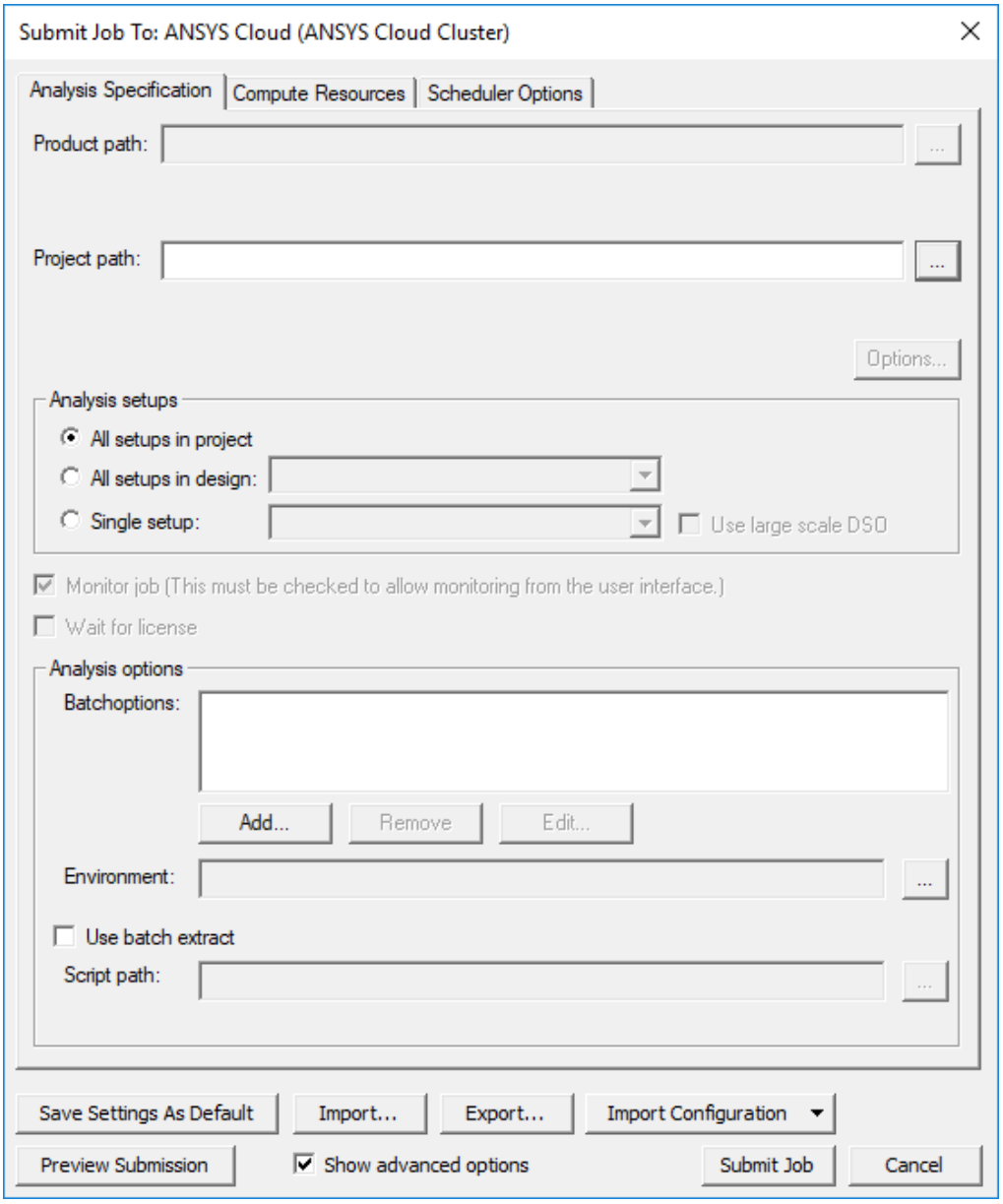
There are two ways that the GUI may be used to submit jobs. The first (and most common) mode requires that the Desktop (UI) process run on a host which is also a submission host for the job scheduler. This mode is called local mode or working mode.

The second mode is only supported on Linux in the Ansys Electromagnetics Suite. In the second mode, an administrator configures the RSM Service to act as an interface to the job scheduler, and starts the RSM Service on a submission host for the cluster. The user runs the Desktop (UI) process on another host (which may be called the post-processing host). To submit a job, the user specifies the host where the RSM Service is running, and the Desktop process connects to the RSM Service over the network to submit the job. In this mode, some configuration is required, and the RSM Service typically must run as a privileged user (for example, root), so that it can launch processes as any user. This mode is useful for cases in which the submission hosts are not able to run graphical processes.

To submit a job:

1. Prepare your design.
2. Open the **Submit Job To** window one of the following ways:
    - On the **Simulation** tab, click the **Submit** icon.
    - Select **Tools** > **Job Management** > **Submit Job**.
    - Select **Maxwell 2D** or **Maxwell 3D** > **Submit Job**.
    - Right-click a solution setup and select **Submit Job**. *In this case, the information in the **Submit Job To** window will be pre-populated from the setup*.
    - From a Command window, use the -showsubmitjob command.

    The **Submit Job To** window appears. The window header indicates your selected scheduler.



This window contains up to three tabs:

- **Analysis Specification** – allows you to specify the product path, project name, setup and analysis options, batchoptions, and environment variables (for advanced users).
- **Compute Resources** – allows you to specify whether to use automatic settings, and to set resource selection and job distribution parameters.
- **Scheduler Options** – allows you to specify the job name and priority. *This tab does not appear if you have selected local RSM as the scheduler*.

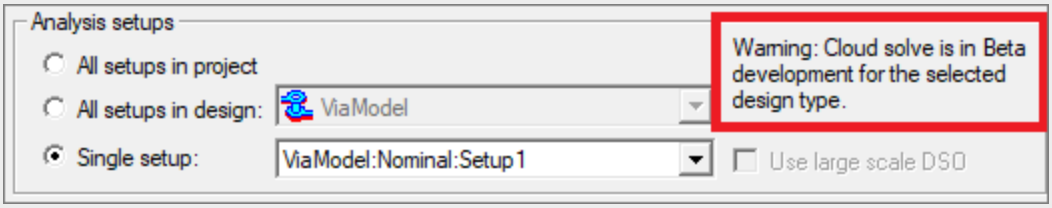
3. On the **Analysis Specification** tab, specify your desired options.

   The **Product path** and **Project** fields support mapped drives. Click the ellipses (**...**) to select files.

   The project can be an archive. The project file pathname must be a UNC path that is accessible from each compute host used for Ansys Electromagnetics jobs. After clicking the **Project path** field's ellipsis button (**...**), a check box allows you to **Use converted UNC path if mapped drive specified**. If you select a project or product on a mapped drive, and check the option, the converted UNC path equivalent to the mapped drive pathname is used.

   Note:

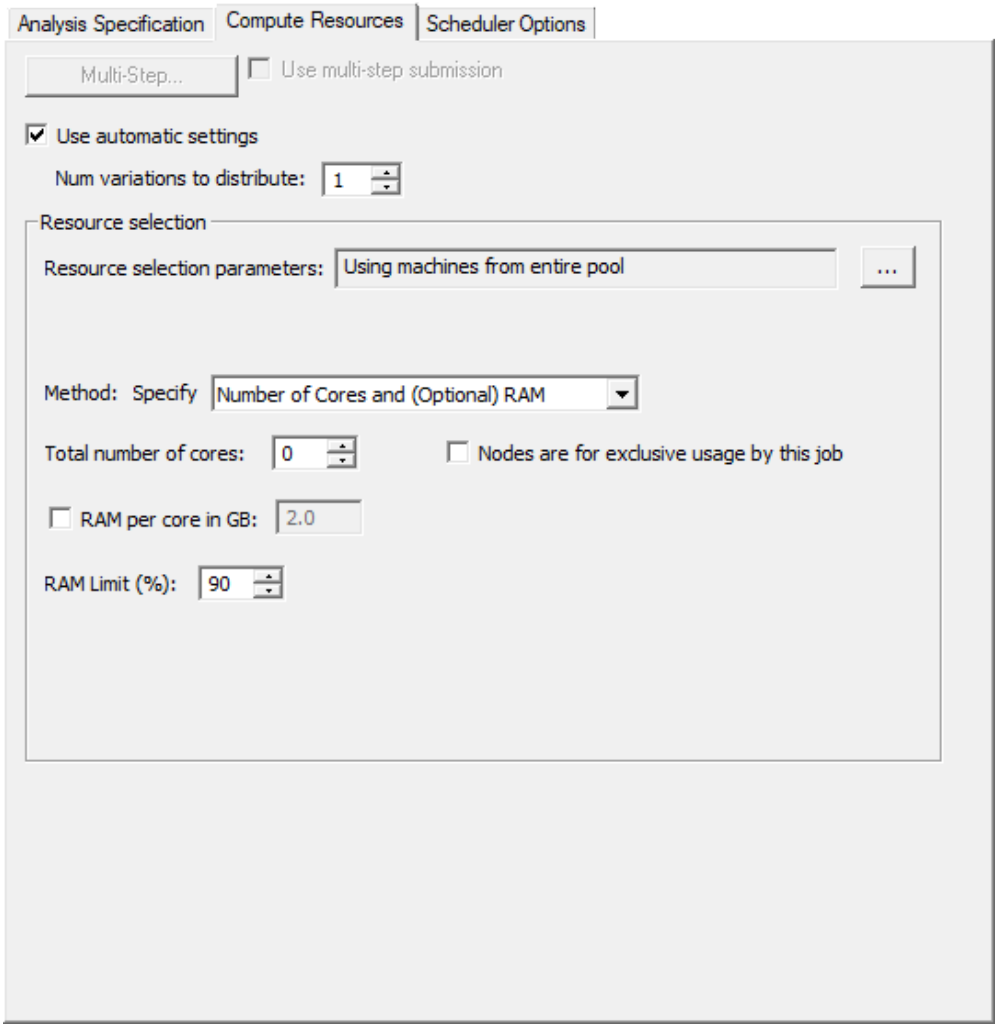
   If you select Ansys Cloud as your scheduler for a design type that is in Beta, you will receive a warning:

   In the **Analysis Setups** area, select the radio button for **All setups in project**, **All setups in design**, or a **single setup**.

   For Parametric setups, you have the option to select **Use Large Scale DSO**. See: Large Scale DSO for Parametric Analysis.

4. Select the **Compute Resources** tab.



Options may vary slightly depending on your selected scheduler.

5. If applicable, determine whether or not to **Use multi-step submission**.
6. Determine whether or not to **Use automatic settings**.
7. Use the buttons in the **Resource selection** area to allocate resources. See: Job Management for Large Scale DSO.
8. If you opted not to use automatic settings, you can also set **Job distribution** parameters. See: Distributed Analysis.
9. If applicable, select the **Scheduler Options** tab and set the **Job name** and **Priority**.

10. If desired, import or export job configurations.
11. If desired, click **Save Settings as Default** to save the current settings and overwrite defaults. These settings are saved on a per-scheduler basis.
12. If desired, click **Show advanced options** to see options for advanced users. See: Running Electronics Desktop from a Command Line.
13. If desired, click **Preview Submission** to view the commands to be sent to the scheduler. Ansys Cloud submissions contain queue (pool) configuration details, including the job's hourly cost in Ansys Elastic Units (AEUs).

    The preview text can be copied to the clipboard.
14. Click **Submit Job** to submit the job to your selected scheduler.

Related Topics:

High Performance Computing (HPC) Integration

Running Ansys Electronics Desktop from a Command Line

Setting HPC and Analysis Options

Distributed Analysis

Job Management for Large Scale DSO

Selecting a Scheduler

Monitoring Jobs

Using Advanced Job Submission Options

Using Advanced Job Submission Options

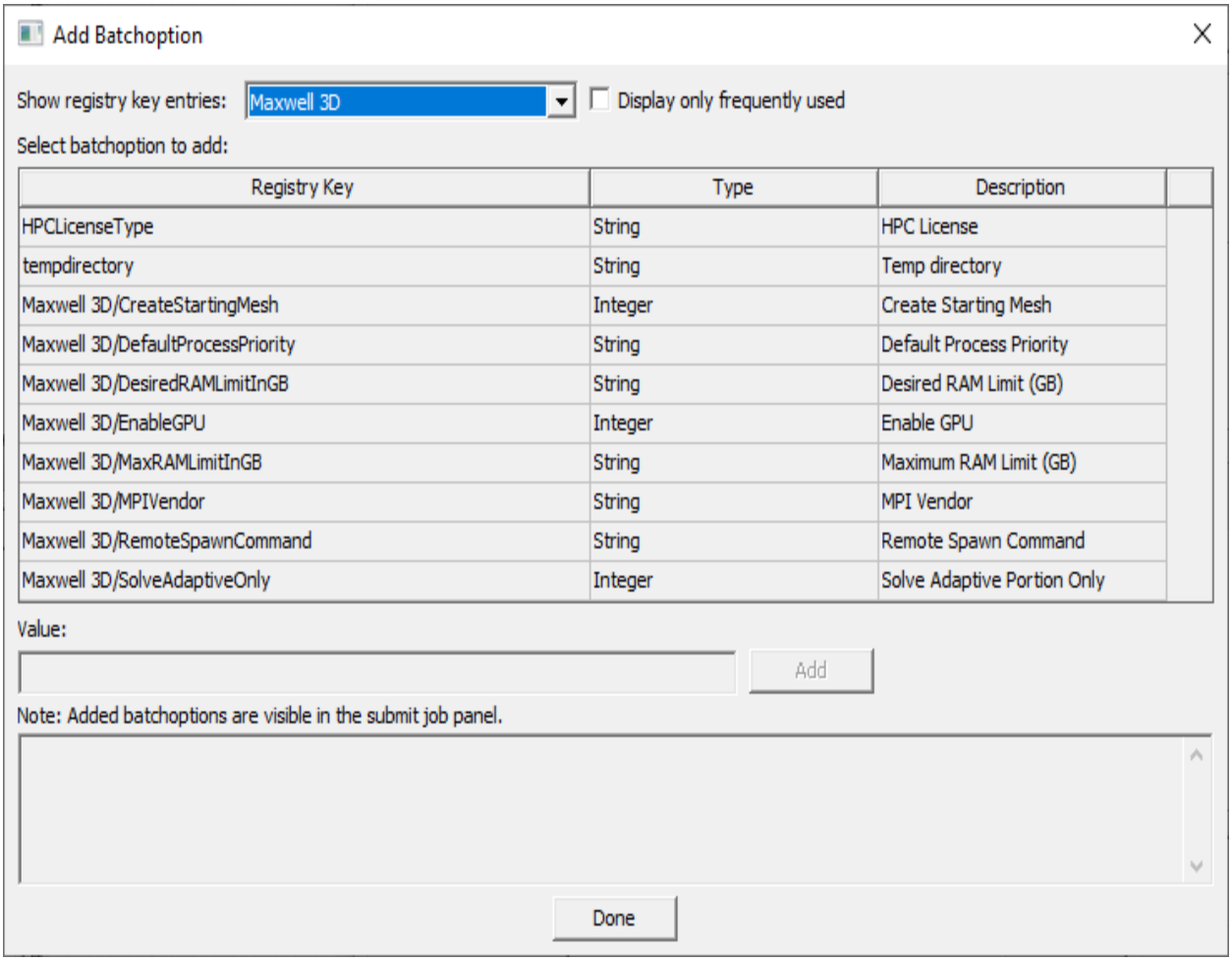
From the job submission window, you can select **Show advanced options** to enable additional analysis options and job submission options.

This topic covers the following:

- Batchoptions
- Environment Variables
- Batch Extract
- Customize Job Submission

Batchoptions

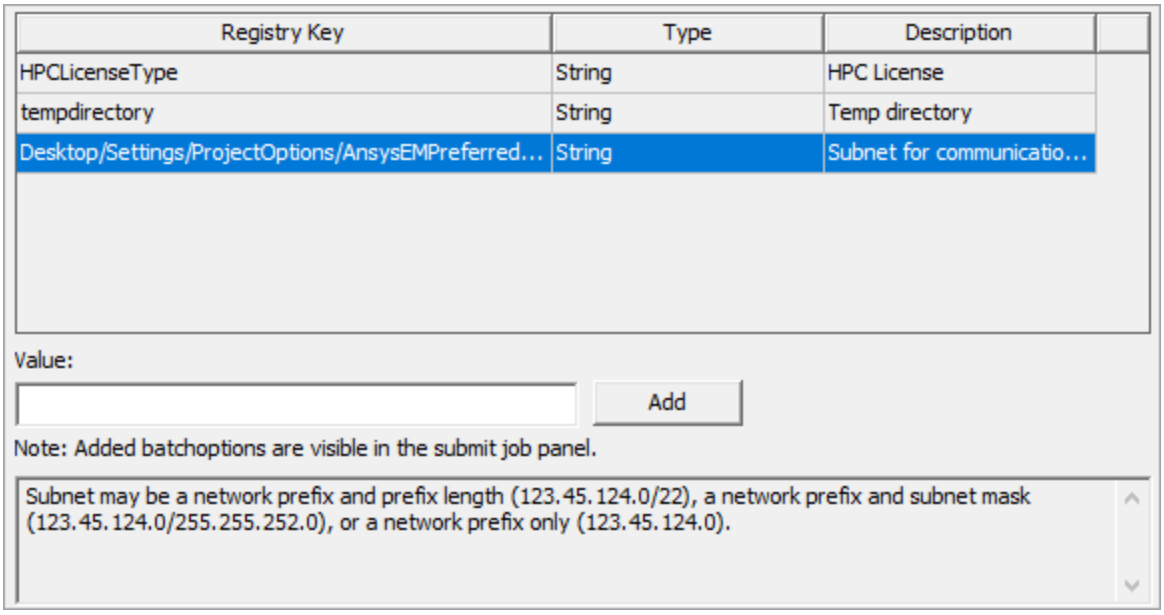
In the **Submit Job To** window, under **Batchoptions**, click **Add** to open the **Add Batchoption** window.



The **Show registry key entries** drop-down menu allows you to select categories of registry keys to display.

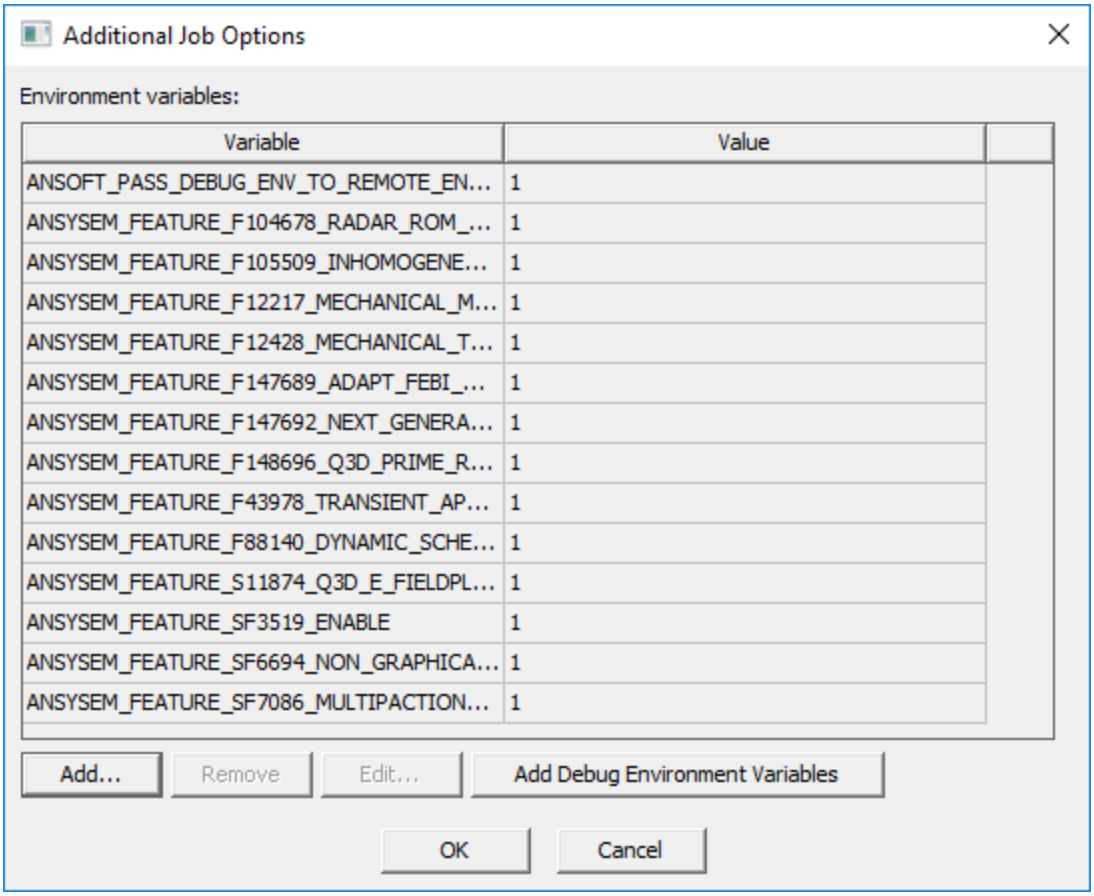
Note that **Display only frequently used** is selected by default. Deselect this to view all options for the selected category.

Select a registry key to activate the **Value** field, where you can enter a value. Selecting a key also populates the bottom of the window with a description of that key:



Environment Variables

In the **Submit Job To** window, under **Environment**, click the ellipses (**...**) to open the **Additional Job Options** window.



This window lists all currently activated environment variables.

Active variables display a value of 1. Inactive variables display a value of 0. Non-binary variables may contain project paths, integer values, etc.

You can perform the following actions:

- **Add** – Add an environment variable. You will be prompted to enter the variable name and value.
- **Remove** – Select and remove an environment variable from the list.
- **Edit** – Select an environment variable and enter a new value.
- **Add Debug Environment Variables** – Activates a selection of debugging variables and adds them to the list. The log files created are only useful to development and if a customer or an application engineer needs to set these environment variables they should be working with a developer directly or indirectly who will know what needs to be set.

Batch Extract

In the **Submit Job To** window, select **Use batch extract** to enable the **Script path** field. Click the ellipses (**...**) to browse and select a VBscript or Python script to execute along with the job.
See: Running Ansys Electronics Desktop from the Command Line for a description of Batch Extract.

Customize Job Submission

In the **Submit Job To** window's **Scheduler Options** tab, advanced options for some schedulers allow you to **Customize job submission**. When the **Override job submission** radio button is selected, user-specified options replace most of the job submission options. When the **Additional job submission options** radio button is selected, user-specified options are appended to the bsub command.

Related Topics:

Submitting a Job

Monitoring Jobs

Running Ansys Electronics Desktop from the Command Line

Using the Command Line to Submit HPC Jobs

Ansys Electronics Desktop can be run from the command line. When using the command line to perform HPC jobs, take the following into consideration.

Distributed Jobs

An Ansys EM batch job which distributes the analysis over several hosts may also be called a distributed job. To submit a distributed job, the following Ansys EM desktop command line options should be used:

- The -Distributed option should be present, and the -Local option should be absent. When running as a batch job under one of the schedulers with direct integration, this option is a directive to the job to 1) obtain the list of hosts allocated to the job, directly from the scheduler, and to 2) use the scheduler to launch the analysis engines on the hosts allocated to the job. The -Distributed option may now have additional options, such as includetypes=xxx, excludetypes=xxx, maxlevels=n, and numlevel1=n, where n indicates and integer, and xxx indicates a list of distribution types or "default".
- The -Machinelist num=*num_distributed_engines* option must be included, where *num_distributed_engines* is the total number of analysis engines to be started on the hosts assigned to the job.

Other examples:

- "Serial Job on a Single Processor" below
- "Distributed Job Using Four Processors" on the facing page
- "Multiprocessing Job Using Four Cores" on the facing page
- "Distributed Analysis and Multi-Processing in the Same Job" on the facing page

Serial Job on a Single Processor

Suppose Ansys Electronics Desktop is installed at "C:\Program Files\AnsysEM\v231\Win64\ and you are using RSM for DSO:

```
C:\Program Files\AnsysEM\v231\win64\ansysedt.exe -ng -
BatchSolve -machinelist num=2
-monitor \\shared_drive\projs\OptimTee.aedt
```

User is using LSF for remote-analysis/DSO

```
bsub -n 1 C:\Program Files\AnsysEM\v231\win64\ansysedt.exe -ng
-BatchSolve -machinelist num=3 -monitor -local \\shared_
drive\projs\OptimTee.aedt
```

Distributed Job Using Four Processors

Ansoft RSM

```
C:\Program Files\AnsysEM\v231\win64\ansysedt.exe -ng -
Batchsolve -monitor -Distributed
-machinelist list="10.1.1.221, 10.1.1.222, 10.1.1.223,
10.1.1.224" \\shared_drive\projs\OptimTee.aedt
```

LSF

```
bsub -n 4 C:\Program Files\AnsysEM\v231\win64\ansysedt.exe -ng
-Batchsolve -monitor
-Distributed -machinelist num=4
\\shared_drive\projs\OptimTee.aedt
```

Multiprocessing Job Using Four Cores

Multi-processing job using 4 cores

```
bsub -n 4 -R "span[ptile=4]" C:\Program
Files\AnsysEM\v231\win64\ansysedt.exe -ng -monitor
-Local -BatchSolve -machinelist num=4 -batchoptions \\shared_
drive\registry.txt \\shared_drive\projs\OptimTee.aedt
```

This requests 4 cores to come from the same machine, as multi-processing needs cores to be on the same machine

Distributed Analysis and Multi-Processing in the Same Job

Distributed-processing using 4 engines and multi-processing using 4 cores, using a total of 16 cores

```
bsub -n 16 -R "span[ptile=4]" c:Program
Files\AnsysEM\v231\win64\ansysedt.exe -ng
-BatchSolve -Distributed -machinelist num=4
-batchoptions \\shared_drive\registry.txt
\\shared_drive\projs\OptimTee.aedt
```

Related Topics:

Running Ansys Electronics Desktop from a Command Line

Integration with Platform's Load Sharing Facility (LSF)

What a Scheduler Does

Installation of Ansys Electromagnetics Suite 2023 R1 on LSF Cluster

Integration of Ansys Electromagnetics products with LSF

LSF Known Issues and Workarounds

LSF Troubleshooting

Aborting an Analysis

Job Import and Export

The bottoms of the scheduler selection and job submission window contain buttons for **Import...**, **Export...**, and **Import Configuration**. Import and Export may be used to save and then restore a frequently used collection of job submission settings, to save multiple sets of settings, or to transfer settings from one machine to another. Electronics Desktop uses Ansoft Electronics Registry Settings (*.areg) files for these purposes:

- **Export...** – exports most of the settings of this window (all tabs) to an *.areg file.
- **Import...** – updates most of the settings in this window (all tabs) from an *.areg file.
- **Import Configuration** – updates DSO settings in this window from any DSO configuration, as shown in the **Configurations** and **Options** tabs of **HPC and Analysis Options**.

> **Important:**
>
> The Design Type of the DSO configuration must match the design type of one of the designs in the project, so the **Project path** must be specified before using the **Import Configuration** button.

Scripting

The SubmitJob scripting command uses job submission settings that have been exported from the **Submit Job** window to an *.areg file. The path to this *.areg file is the first argument to the SubmitJob scripting command. See: Job Submission Scripting.

Multi-Step Job Submission

Multi-step job submission allows you to divide the simulation of a project on a cluster as two or more jobs, each of which has unique resource specifications. For example, creating the initial mesh and doing adaptive refinement can use a single machine, while frequency sweeps can easily be distributed over many machines. Breaking up the simulation into multiple steps allows the first job to do initial meshing and adaptive passes, only reserving a single compute node, or maybe even reserving just a partial node. The second job can then do the frequency sweep(s),

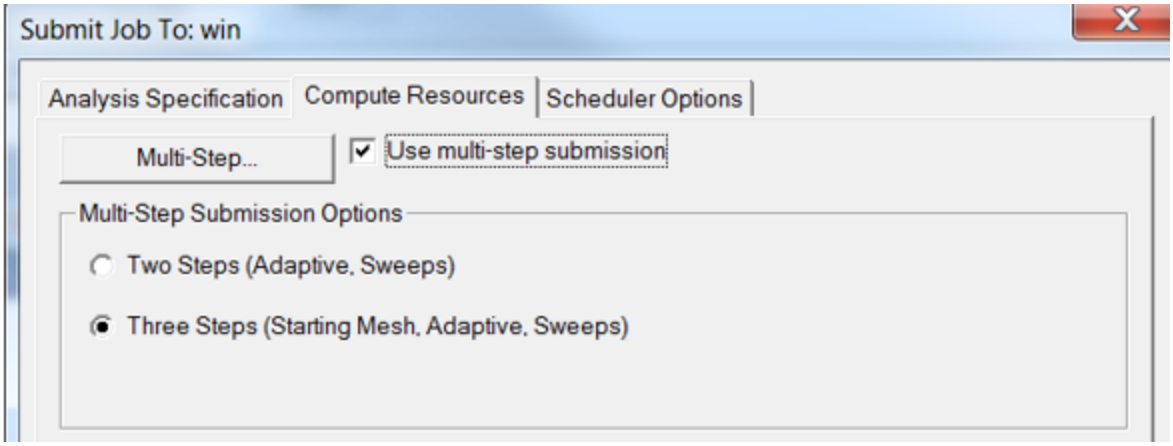
reserving and using multiple nodes. Note that while the first job runs, because it may only be using one node, other nodes are available for other jobs. The Ansys Electronics Desktop job submission GUI allows you to submit multi-step jobs, and specify compute resources individually for each step. Electronics Desktop can also be used to monitor the execution of multi-step jobs.

Limitations

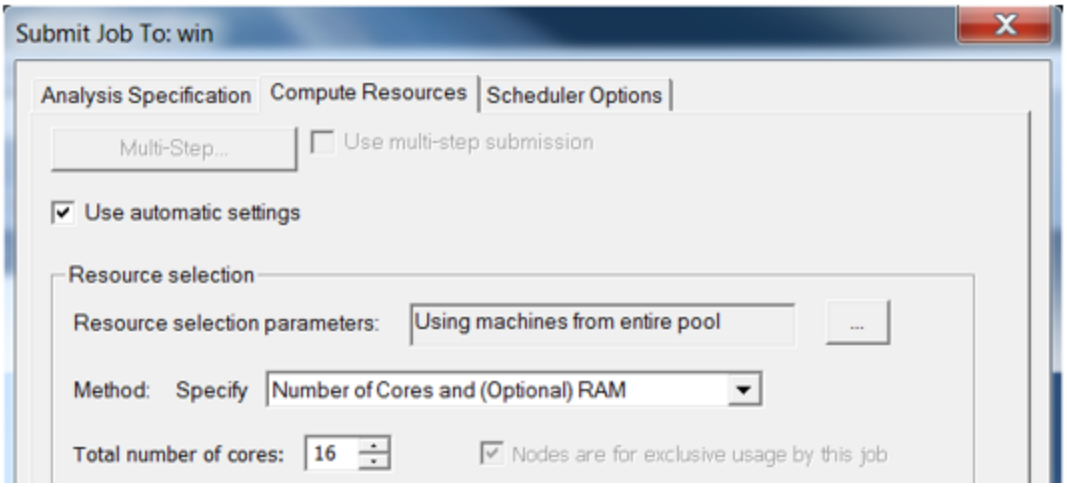
Multi-step job submission is only allowed for a single setup of a project. Only certain design types (or setup types for a given design type) offer this functionality. Ansoft RSM does not support multi-step jobs because it does not have queuing capabilities.

How-to Specify Multi-Step Job Submission

From an open project, right-click a setup in the project tree, and select **Submit Job...** on the shortcut menu. This pre-populates the **Submit Job To** window's **Analysis Specification** tab for the selected setup. Select the **Compute Resources** tab. If the design setup and selected scheduler allow for multi-step submission, and your computing resource supports it, the **Compute Resources** tab shows the **Multi-Step...** button and the **Use multi-step submission** check box is enabled.



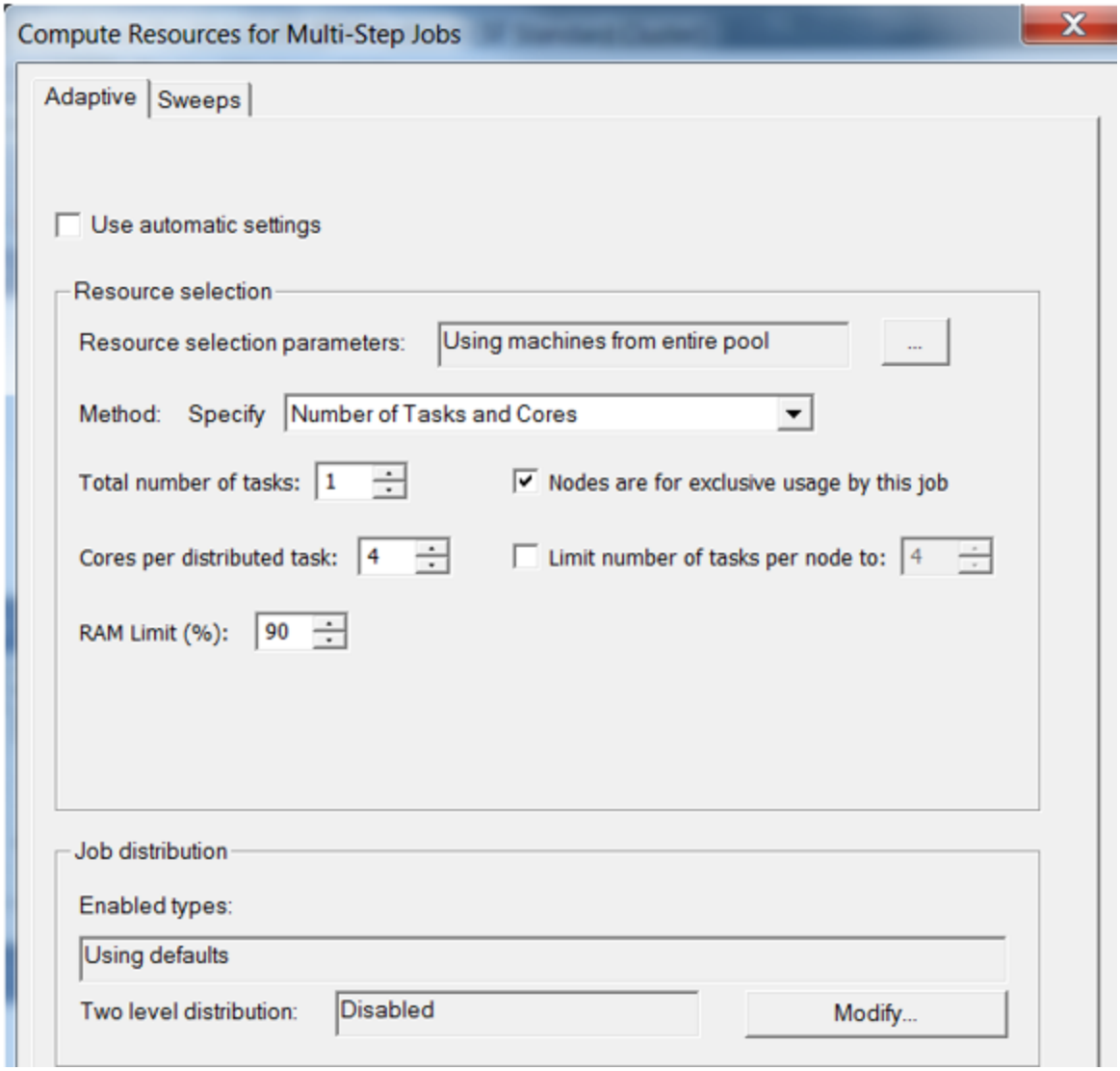
If the Multi-Step button and check box are not enabled it could be because you have not selected a single setup, the design type of the setup does not support Multi-Step, or the scheduler type (e.g., RSM) does not support it.



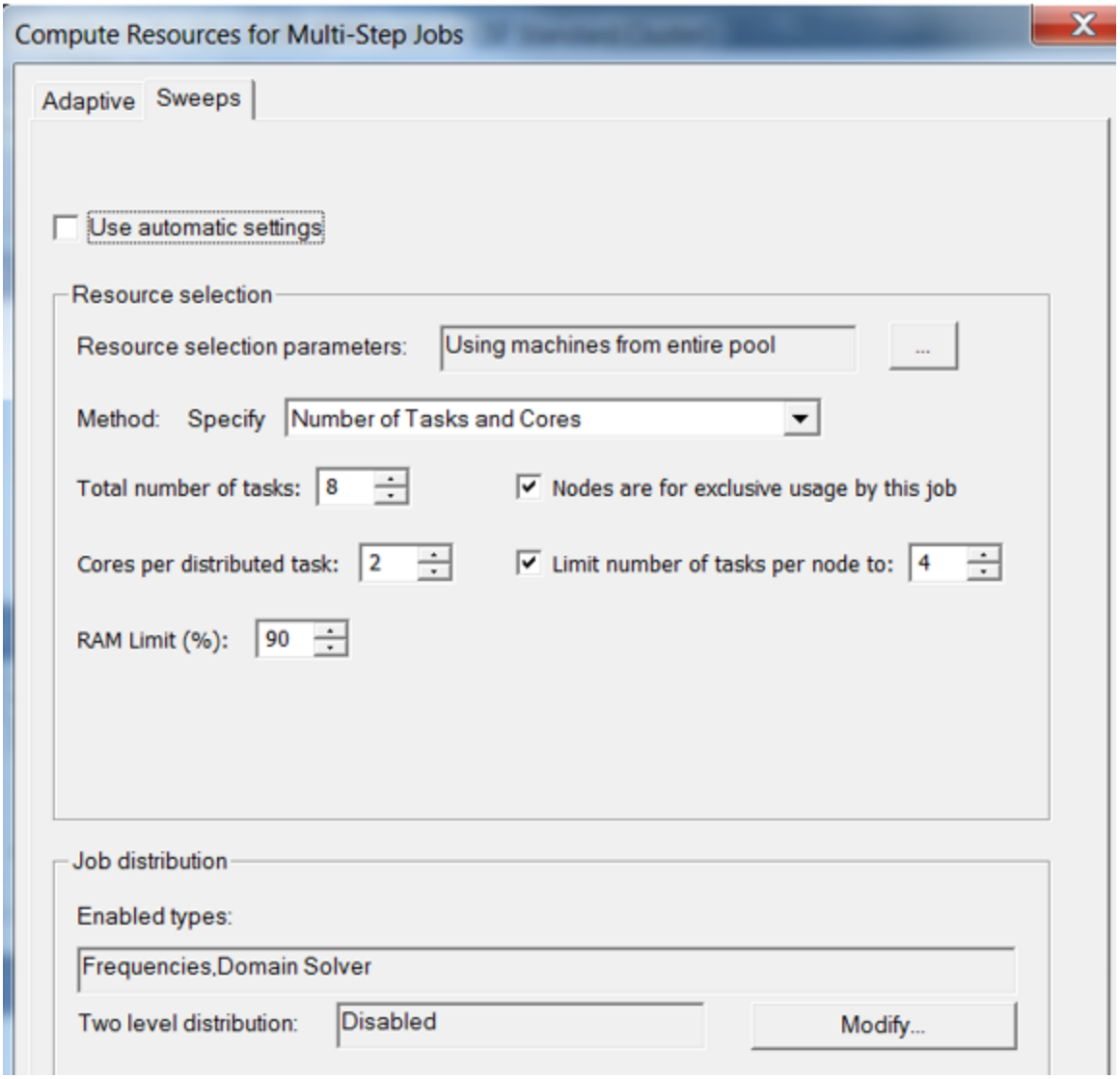
On the **Compute Resources** tab, check the **Use multi-step submission** box and select the appropriate submission option, that is:

- Two Steps for Adaptive and Sweeps
- Three Steps for Mesh, Adaptive, and Sweeps

Then click **Multi-Step…**, which will bring up a **Compute Resources for Multi-Step Jobs** dialog box where there is a tab for each step used, that is for potential for Mesh, Adaptive, and Sweeps.

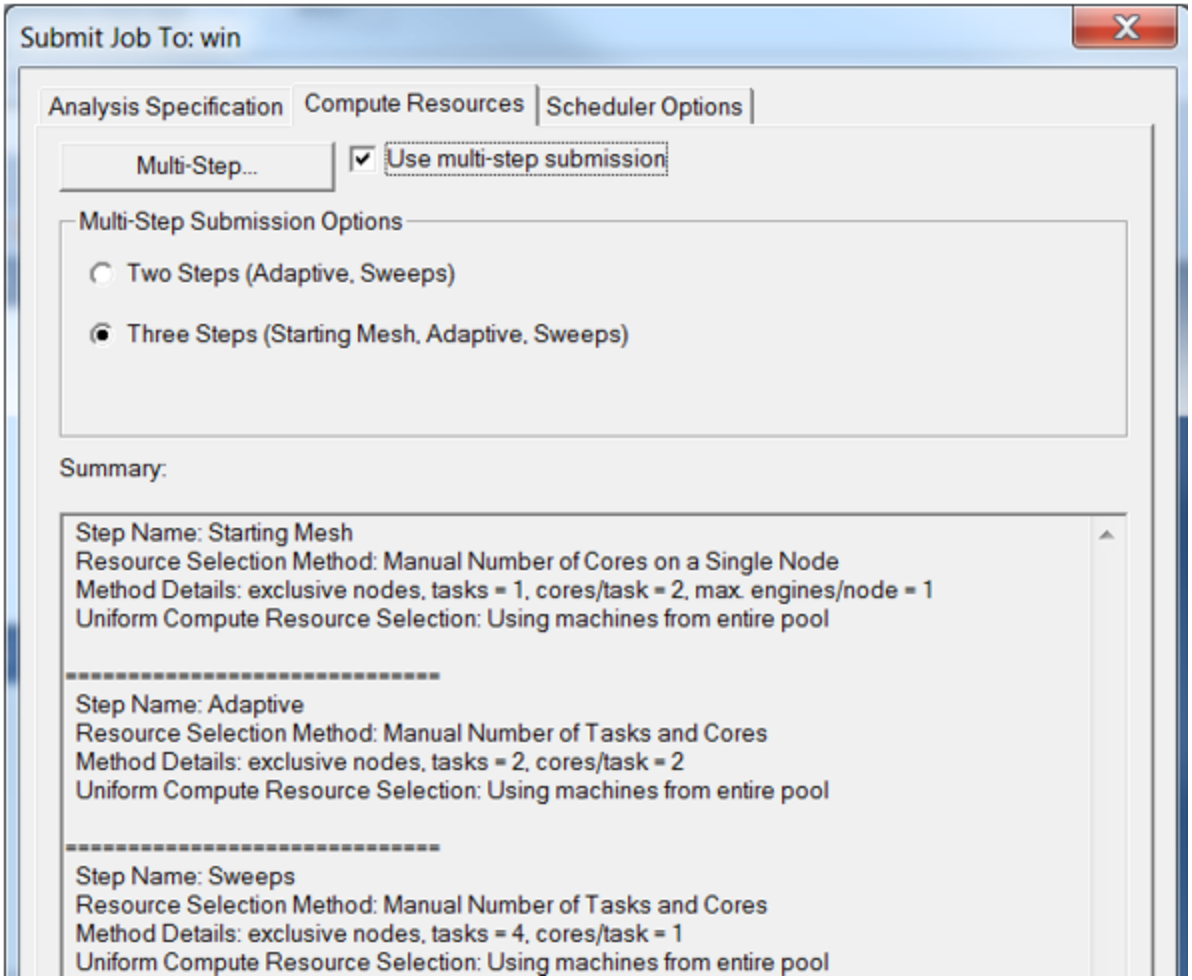


Notice, for example, that the **Sweeps** tab lists the same resources choices but they can be assigned differently.

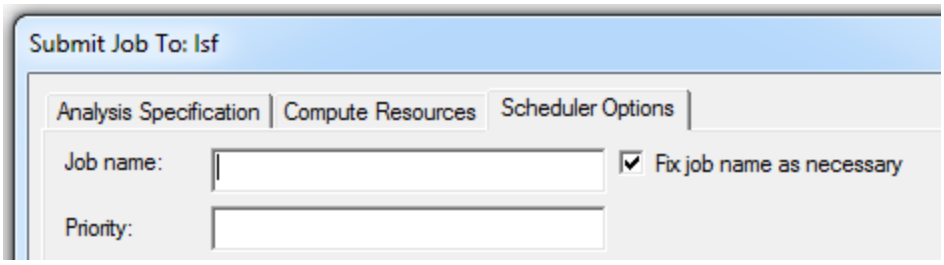


Set the desired compute resources for each step and click **OK**.

The Summary field of the **Compute Resources** tab shows a text summary of resource specifications used for each step.

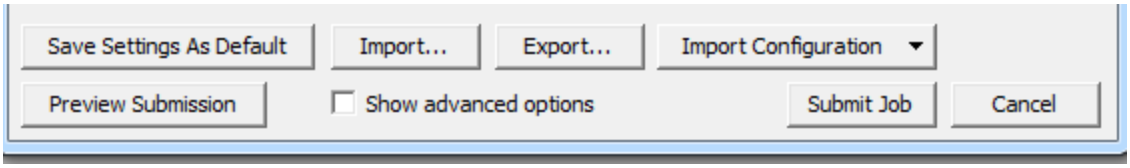


If desired, set the job name on the **Scheduler Options** tab. When the jobs are submitted, each will have a name given by job name (if any) appended with the name of the step for the job. For example, if the job name entered is “MultiStep”, then the individual jobs might be named “MultiStep-Adaptive” and “MultiStep-Sweeps”.



There is a new **Fix job name as necessary** check box for the job name. This applies only to certain schedulers where there are constraints on the job name. It is useful to ensure proper submission in the case where appending the step name results in an invalid job name.

When you click **Preview Submission**, you see a preview for each step, all in the same output window. Any errors or warnings for a step, generated during preview validation, are displayed with the text for the preview of that step.



When you click **Submit Job**, each job is submitted individually, and dependencies are set with the scheduler so that subsequent jobs wait for the prior step's job to complete before starting. Any errors in job submission for any step prevent further steps from being submitted.

The dialog reporting successful submission has been modified slightly for multi-step jobs. It will show the jobs IDs of all jobs that are part of the multi-step job sequence.

Multi-Step Job Monitoring

You can monitor the job step in progress. When one job completes, the status shows "Completed", but monitoring restarts once the next job step is running.

Aborting a Job Step

You can also abort the job step in progress. With multi-step jobs submitted from the Electronics Desktop GUI, this will also cause remaining jobs (which would otherwise remain queued in the scheduler) to be canceled.

Archive Projects for Multi-Step Job Submission

Note that you can submit archive projects. Monitoring is based on the archive for the first step, then on the extracted project for subsequent steps.

Submitting a Job without Opening the Project

You can also submit a job without opening the project. This can be done by choosing **Tools** > **Job Management** > **Submit Job…** and then manually entering the project path (You can also use **Browse** can to select the project.) Note that you must also select a single nominal setup before the **Use multi-step submission** check box is enabled on the **Compute Resources** tab.

Windows to Linux Job Submission

Given a set of prerequisites, Ansys Electronics Desktop can permit Windows to Linux job submission as part of HPC.

Prerequisites for Job Submission

Directory Shared between Windows and Linux

For all jobs submitted to a Linux cluster, the project file is required to be in a directory that is accessible from all execution hosts used by the job. For submission of jobs from a Windows host to a Linux cluster, the project file must also be accessible from the Windows host where the GUI runs. There must be a directory shared with both Windows and Linux hosts, and the project file may be in a subdirectory (at any level) of the shared directory.

Network Access from Windows Host to Linux Job Management Host

The job is submitted to the cluster from a Linux host configured for submission of jobs to the Linux cluster. We call this Linux host the “Job Management” host. The information about the job to be submitted is transmitted to the Job Management host over the network. As a result, the Windows host where the GUI runs must have network access to the Job Management host. If this communication is blocked, then job submission from a Windows host to the Linux cluster will not be possible. Communication could be blocked if there is a firewall or if the Linux cluster is only on a private network, for example.

RSM Service Running on Job Management Host

The ansoftrsmservice must be running on the Linux Job Management host. Before the ansoftrsmservice is started, it must be configured for submission of jobs to the cluster. The SchedulerName and ConfigString fields in the Scheduler block of the ansoftrsmservice.cfg configuration file must be specified. The contents of these fields are described in the following table:

Field Name	Contents	Examples
SchedulerName	Identifier of Scheduler	IBM Spectrum LSF: ’`lsf`’ PBSPro or Torque: ’`pbs`’ Univa, SGE, etc.: ’`sge`’ SLURM: 'generic'
ConfigString	Directory containing scheduler commands	IBM Spectrum LSF: ’’ (not required) PBSPro or Torque: ’`/opt/pbspro/PBSPro_13.0.0/default/bin`’ Unive, SGE, etc.: ’`/opt/univa/bin/lx-amd64`’ SLURM:'{"Proxy":"slurm"}'

The environment should be configured for job submission before starting the ansoftrsmservice. The ansoftrsmservice should be run as a non-privileged user; no special privileges are required. It should be run as a user without login privileges, so that only privileged users have access to this process.

Prerequisites for Job Monitoring

For job monitoring, all prerequisites for job submission are required. One additional requirement, described below, is also required for job monitoring.

Network Access from Windows Host to Linux Cluster Hosts

In order to obtain full monitoring information from a job, the Windows host needs access to some of the job processes. That is, the Windows monitoring host requires network access to the

processes running on the Linux cluster execution hosts. If this communication is blocked, then only limited monitoring information is available.

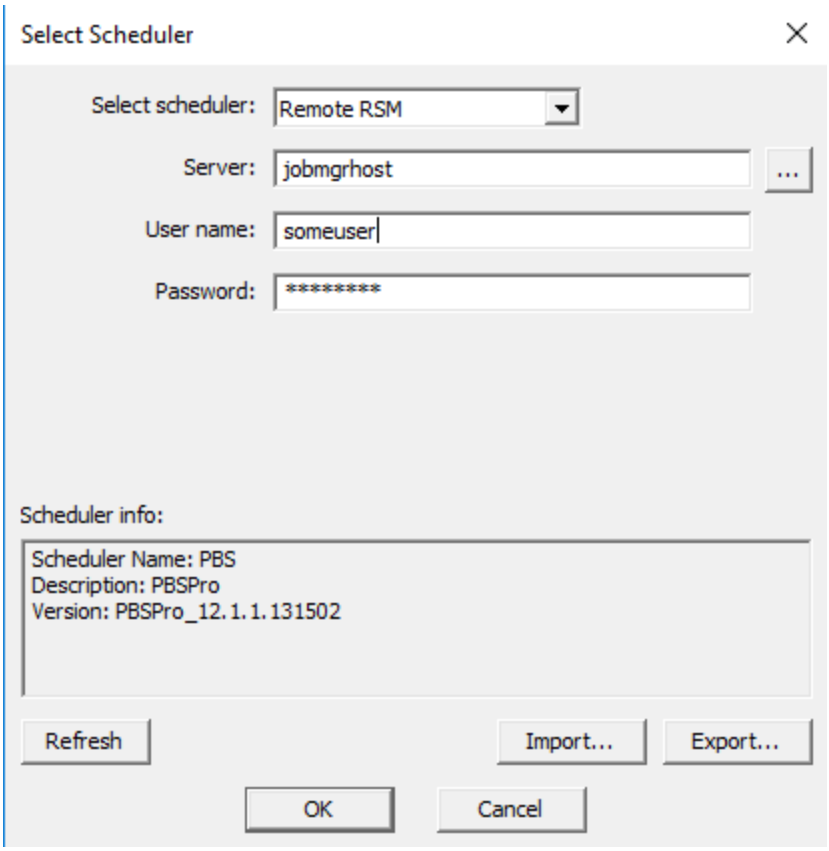
Supported Schedulers

This feature may be used with all Linux schedulers for which job submission from the GUI is supported:

- IBM Spectrum LSF
- Univa Grid Engine (formerly Sun Grid Engine, Oracle Grid Engine)
- PBSPro
- Torque

Select Scheduler Dialog

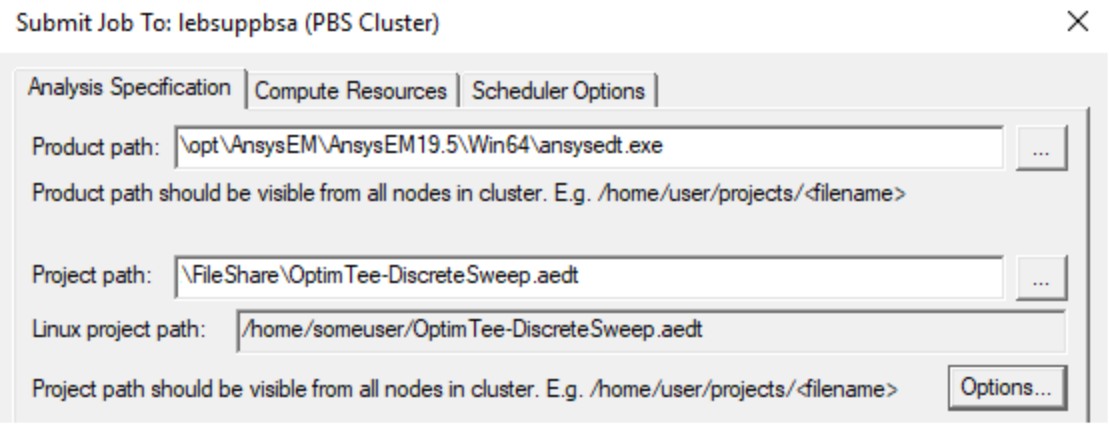
If you select the **Use a computer on the network** option, you can enter a username and password. This username and password are used when the job is submitted to the Linux scheduler.



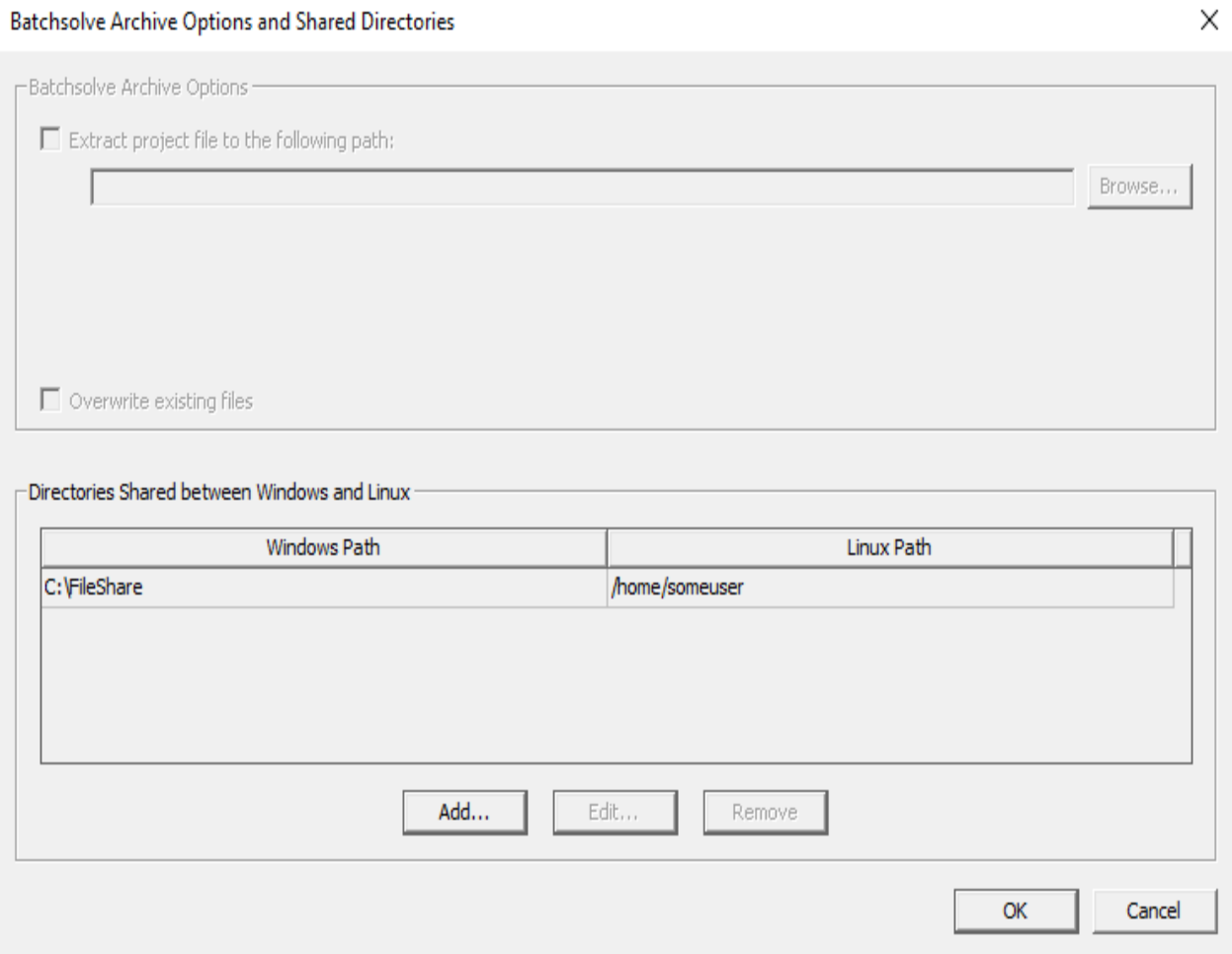
Submit Job Dialog

In the **Submit Job** dialog box, you must enter the Linux path to the product in the Product path edit control. The browse button (labeled "...") may not be used to browse for the product. There is no requirement for the product installation directory to be accessible from the Windows GUI host.

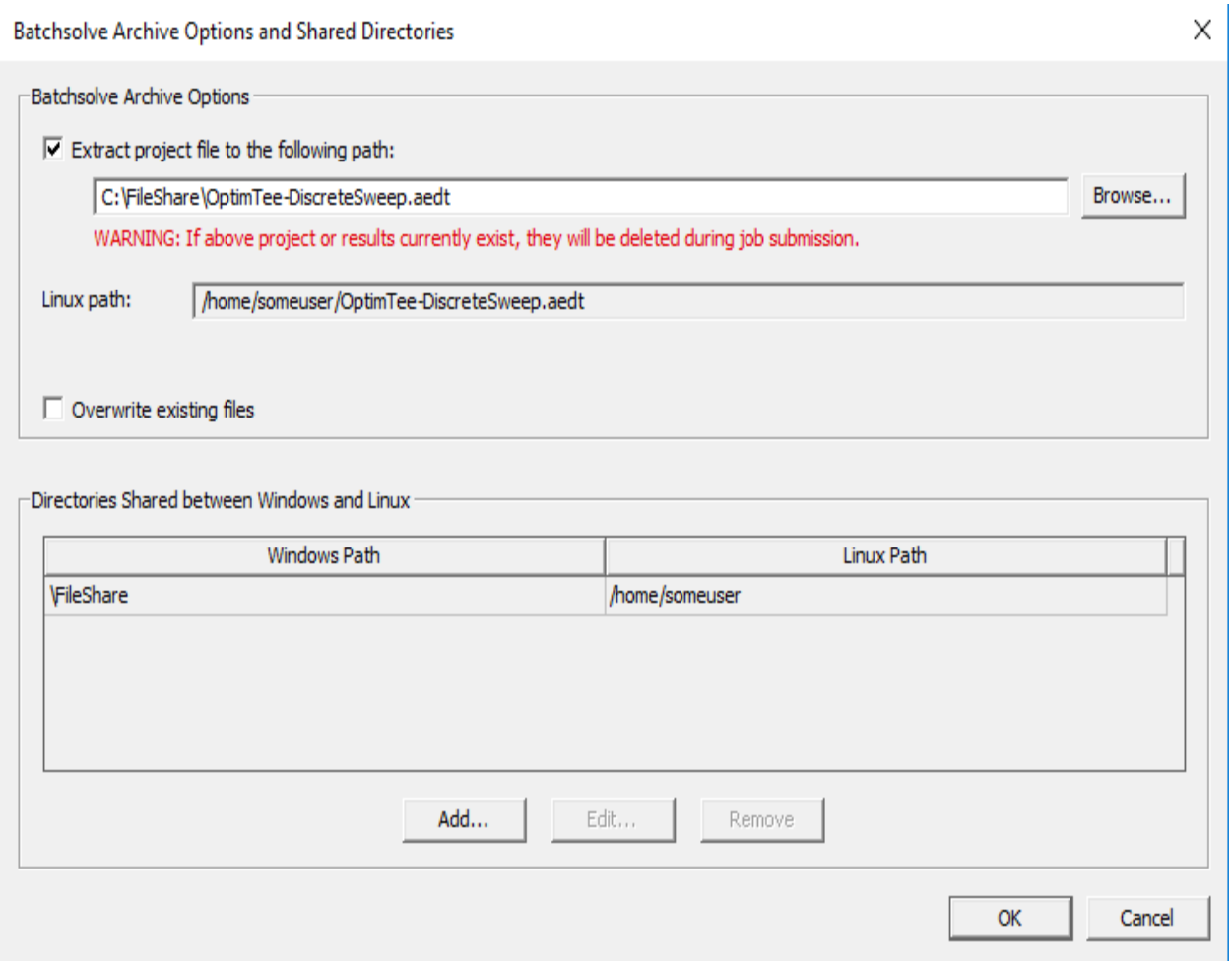
You must enter the Windows path of the project file in the Product path edit control or use the browse button (labeled "...") to select the Windows path of the project file. If the Linux path of the project file can be determined from the specified Windows path and the directories shared between Windows and Linux, then the Linux path of the project file is shown in the Linux project path edit control. This edit control cannot be edited directly.



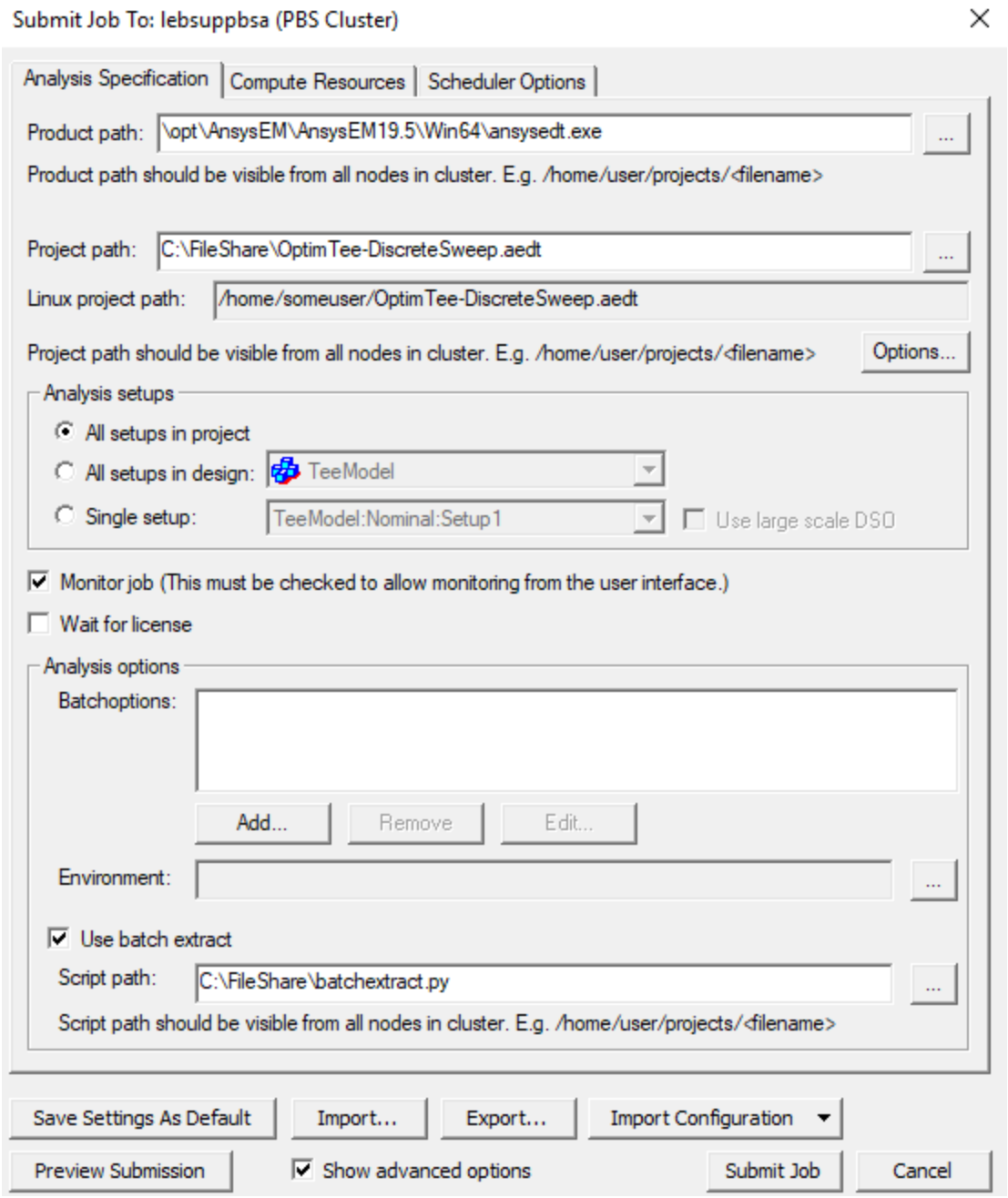
The **Options** button activates a dialog that you can use to specify archive options for a job. The lower portion of this dialog allows you to specify one or more directories shared between Windows and Linux. The mapping of directories between Windows and Linux is shown in a grid which displays the Windows path and the Linux path for each shared directory. There are also buttons to add a new shared directory, to edit an existing shared directory, or to delete one or more shared directories. The Windows path or the Linux path of any shared directory may be selected in the grid and directly edited, as well.



If you specify a project in an archive, the dialog activated by the **Options** button may be used to specify the Windows pathname of the project to be extracted from the archive. If this is done, the Linux pathname of the target project is determined from the directories shared between Windows and Linux and shown in the upper portion of this dialog.



If you specify a batchextract script, the Linux path of the batchextract script is determined from the Windows path of the batchextract script and the directories shared between Windows and Linux.



User Passwords are Encrypted

Jobs are submitted to the Linux cluster using the user name and password entered in the **Select Scheduler** dialog box. These settings are persistent; in general, these settings need to be entered only if they change. To ensure security, user passwords are stored in an encrypted format. When a job is submitted from a Windows host using the ansoftrsmservice running on a Linux submission host, the user credentials are sent over the network in an encrypted format.

Related Topics

Configuring Distributed Analysis

Editing Distributed Machine Configurations

Selecting an Optimal Configuration for Distributed Analysis

Large Scale DSO for Parametric Analysis

High Performance Computing (HPC) Integration

Job Submission Scripting

To help with automation, you can submit batch jobs can through script commands of the **oDesktop** object. The **SubmitJob** script command uses job submission settings that have been exported from the **Submit Job** dialog box to a .areg file. The path to this .areg file is thus the first argument to the SubmitJob command. Additional arguments include the path to the project file, the design name (if restricting the solve to a particular design), and the setup name (if further restricting the solve to a single setup within a design).

For further automation, you can use the SelectScheduler scripting command to determine what scheduler to use for submission, to include options for head node, username, and whether to require password entry from the user. (If the username differs from the cached username, or the force password flag is set, then the **Select Scheduler** dialog box appears.) If there are any issues with the scheduler selection (for example, a password is required or the requested host wasn't found), then the **Select Scheduler** dialog box appears. This is the only part of job submission scripting that may required user intervention. This same mechanism is used if, from within the SubmitJob command, there is failure to connect to the scheduler. Even though there are allowances for graphical user intervention if something goes wrong, if the password (if required) is cached and all settings are correct, the entire submission process can run non-graphically and fully automated.

Limitations

All settings besides the arguments passed to the **SubmitJob** command must be stored in the .areg file containing settings exported from the **Submit Job** dialog box. These include (but aren't limited to) batch options, environment variables, batch extract settings, and compute resource selections. To run many job submission scripts with variation of these settings, there must be multiple .areg files available.

Note that the same project can be submitted multiple times with a single script. Care must be taken in this situation because each time a project is submitted, the state-keeping files used for monitoring are removed so that the job can create them from scratch to ensure consistency. While this ensures proper monitoring for a job that is just being submitted, it could interfere with monitoring (or even correct solving if a lock file is deleted) of a job that is already in progress. Because of this, if the same project is to be re-submitted from within a single script, the job should be monitored (waiting for completion) before trying to submit it again. This monitoring/waiting can be done with a combination of a single **LaunchJobMonitor** command followed by a loop that checks the result of a **RefreshJobMonitor** command.

How to do Job Submission Scripting

The typical scenario for job submission scripting would be to do the following:

1. Manually select the scheduler. Use the **Select Scheduler** dialog box to open the **Submit Job** dialog box.
2. Choose a representative project (with the desired design type), and select appropriate analysis settings.
3. Make the required compute resource selections and try to preview the job.
4. If preview is successful, export the dialog settings and record the path to the new .areg file.
5. Create a script containing at a minimum a **SubmitJob** command with the path to the .areg file, and the path to the project file. Note that there must be double backslashes for each backslash of a path, since the backslash is an escape character. When the script is run, and all is successful, there should be a message in the message windows stating that the job was submitted, including the job ID(s). There could be multiple job IDs if multi-step submission is used.

See the Scripting help (click **Help>Maxwell Scripting Help**) for details on the **SelectScheduler**, **SubmitJob**, **LaunchJobMonitor** and **RefreshJobMonitor** commands.

Related Topics

High Performance Computing (HPC) Integration

Running Ansys Electronics Desktop from a Command Line

Monitoring Jobs

There are a number of tools that allow you to monitor jobs, including the **Monitor Job** window and detailed .log files.

- Monitor Job Window
- Monitoring Ansys Cloud Jobs
- Monitoring Large Scale DSO Jobs

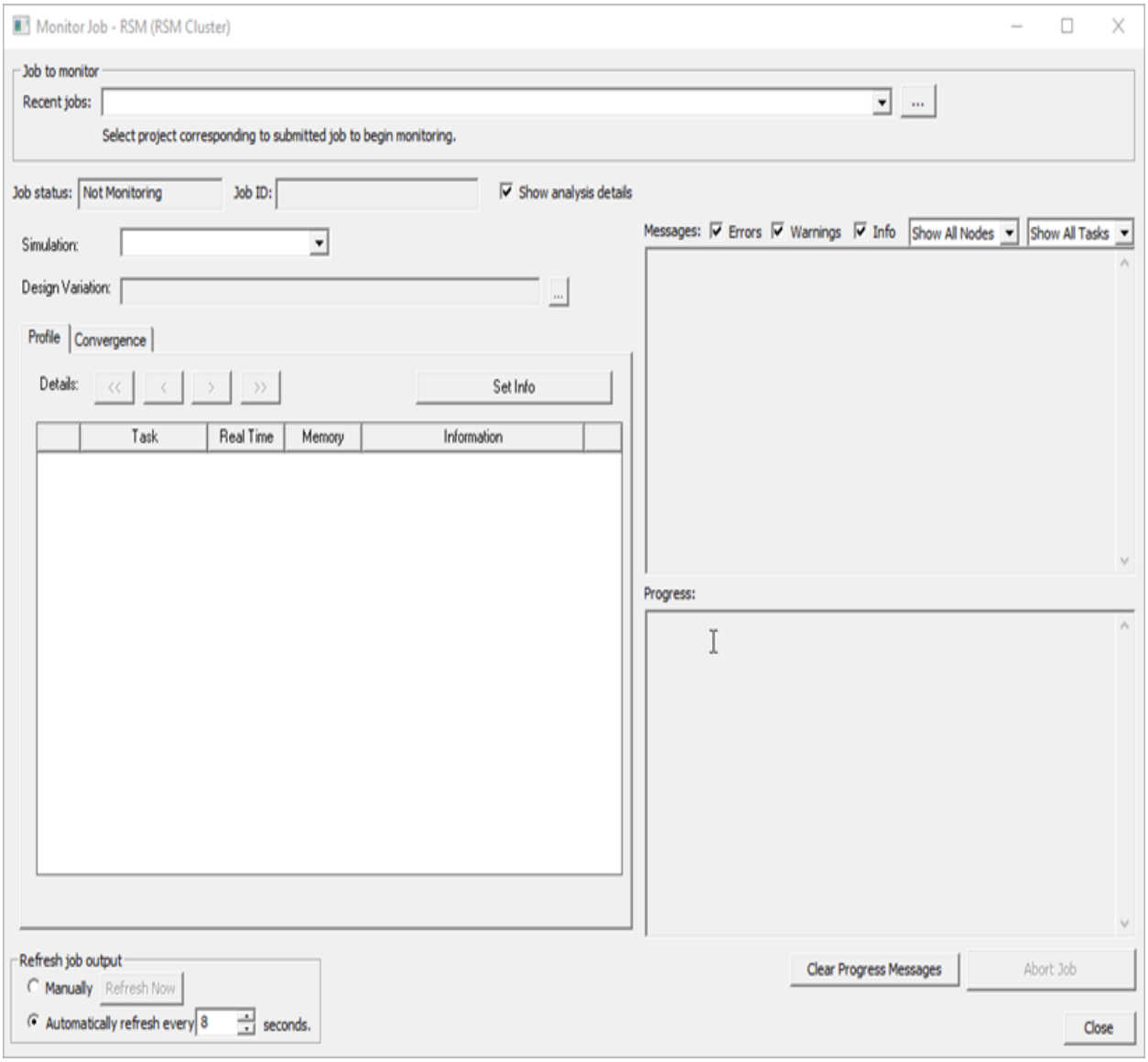
Monitor Job Window

The **Monitor Job** window allows you to monitor the progress and status of jobs, including information on variations solved so far, variations currently solving, and the number of variations remaining.

To monitor jobs:

1. Click **Tools** > **Job Management** > **Monitor Jobs**, or:
   select the **Simulation** tab and click the **Monitor** icon, or
   from a Command window, use the -showmonitorjobs command.

   The **Monitor Job** window appears.



Note:

The window may look slightly different, depending on your selected scheduler.

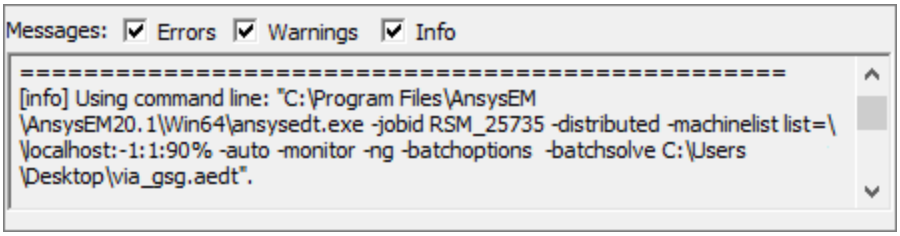
The **Monitor Job** window contains the following areas:

- **Job to Monitor** – Allows you to select either a recent job, or a project containing the job you want to monitor. You can select an archive.
- **Job Status** – The current job status. The normal progression is: Starting Monitoring, Queued, Running, Shutting Down, Completed. A status of Unknown may indicate a connection problem.
- **Job ID** – Identified by the scheduler prefix and job number.

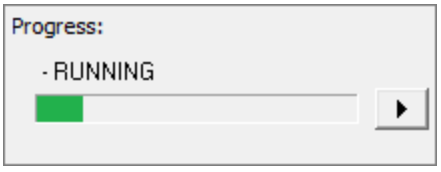
- **Simulation** – Drop-down menu that allows you to select an individual simulation setup.
- **Design Variation** – Click the ellipses button (**...**) to select or deselect design variations.
- **Profile Tab** – Displays detailed information about completed tasks, including execution time and memory usage.

Task	Real Time	CPU Time	Memory	Information
Start				Time: 10/16/2019 14:17:38; Host:: Processor: 12;
				Executing from C:\Program Files\AnsysEM\AnsysEM20.1\Win64\Q
				Performing minimal design validations
InitMesh	00:00:08	00:00:07	106 M	70131 triangles
PreProc	00:00:00	00:00:00	106 M	70131 triangles
AdaptMesh_1	00:00:01	00:00:00	107 M	78514 triangles
Solve(1p)_1	00:00:00	00:00:00	126 M	51207 unknowns
ErrorCalc_1	00:00:00	00:00:00	126 M	78514 triangles
PostProc	00:00:01	00:00:00	126 M	995 elements
Solution Process				Elapsed Time: 00:02:15
Stop				Time: 10/16/2019 15:15:30; Status: Normal Completion

- **Convergence Tab** – Displays the completed, maximum, and minimum **Number of Passes**.
- **Messages** – Displays errors, warnings, and job information. Use the check boxes to choose which of these to display.



- **Progress** – Displays progress bars when tasks are currently in progress.



  The arrow button opens a menu that allows you to either **Abort** that analysis, or perform a **Clean Stop**.
- **Refresh Job Output** – Allows you to select either manual or automatic refresh for the **Monitor Job** window.

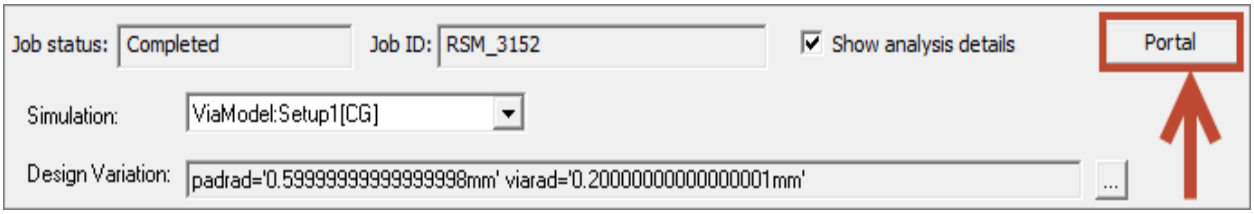
To display less information, deselect **Show analysis details**. This removes the **Simulation**, **Design Variation**, **Profile**, and **Convergence** information, instead displaying only **Messages** and **Progress**.

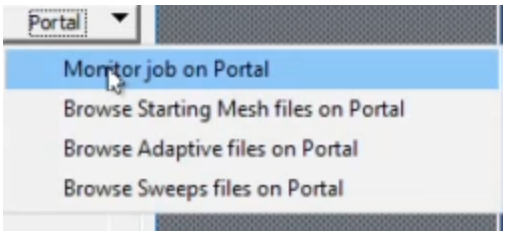
1. Click **Close** when you are done monitoring.

Monitoring Ansys Cloud Jobs

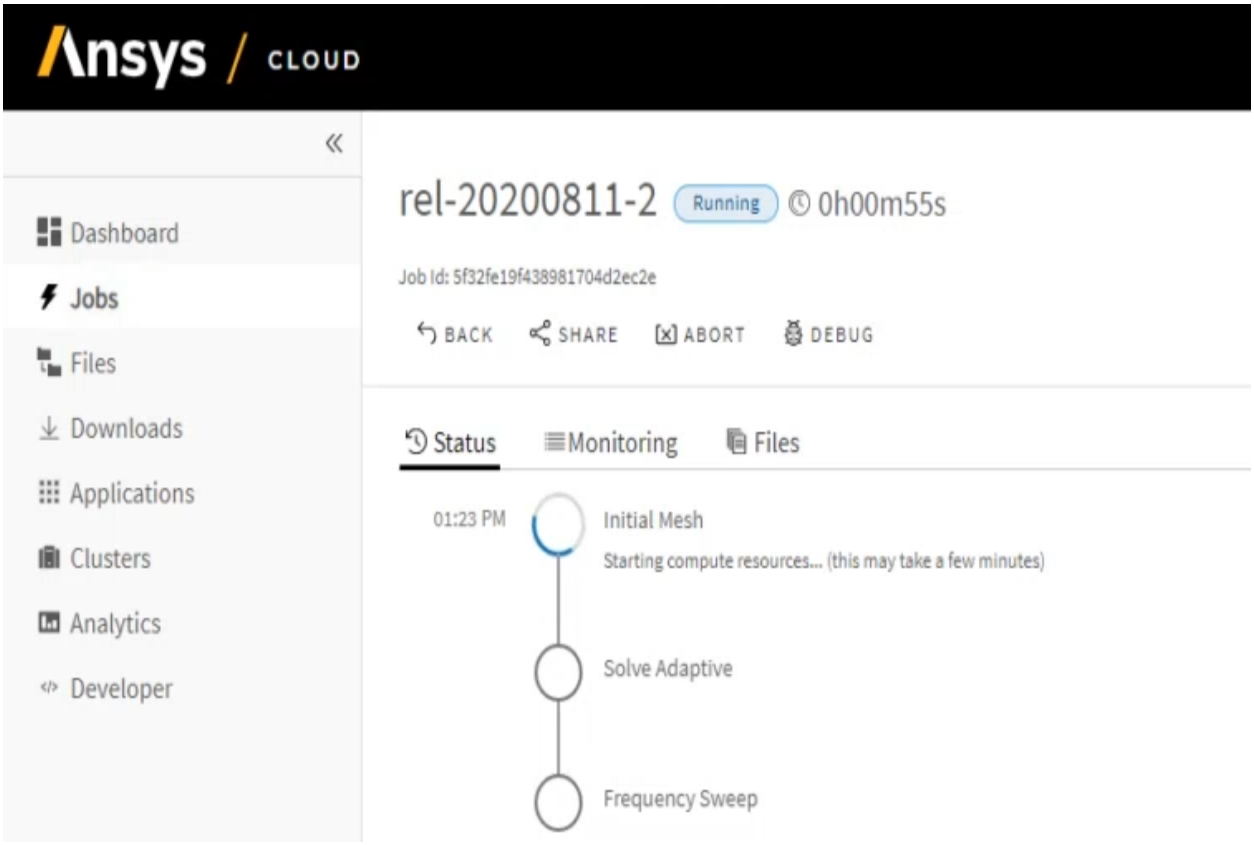
For Ansys Cloud submissions, the **Monitor Job** window includes two additional items:

- **Portal** – link to the Ansys Cloud Portal, where job details and additional monitoring information are available.

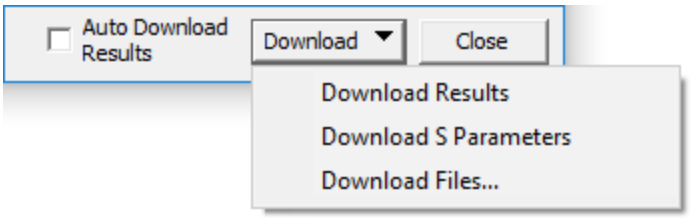




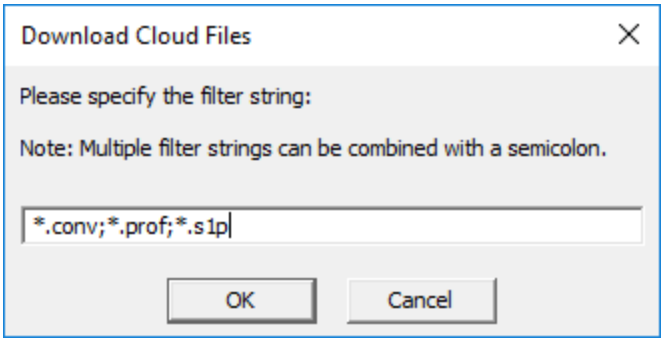
Monitor job on Portal Opens a Portal Web interface that lets you monitor an Ansys Cloud job. The additional menu options for **Portal** are for multi-step submissions, opening the Portal Web interface directly to folders with Starting Mesh Files, Adaptive Files, or Sweeps files.



- **Download** – button that allows you to download the results, s-parameters, or specified files. These are saved in a folder with same name as the job ID inside the `/<project-name>.aedtdownload` folder (or `/<project-name>.aedtzdownload` folder, if the submission was based on an archive). You can also select the **Auto Download Results** check box to have the results downloaded to this location automatically.



If you choose to **Download Files**, the **Download Cloud Files** dialog box appears:



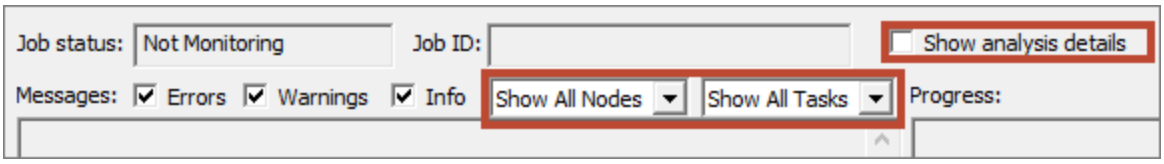
Specify the types of files you want to download (separated by semicolons) and click **OK**.

Monitoring Large Scale DSO Jobs

For Large Scale DSO submissions, the Monitor Job window includes the following items:

- **Show Messages in Node Tree** – Instead of viewing job details, you can select this option to view nodes and tasks in a tree format.

Drop-down boxes allow you to filter which Nodes and Tasks to display in the tree.



To leave this mode, select **Show analysis details**.

Ansys Cloud Support for HPC Job Management

Ansys Electromagnetics Desktop supports the Ansys Cloud service that allows quick and easy access to cloud-based HPC resources. The service is accessible through both the Simulation Ribbon and **Tools** > **Job Management**.

Note	For this release you can submit HFSS, HFSS 3D Layout, Icepak, Maxwell 3D, Maxwell 2D, Mechanical, Q3D Extractor, and 2D Extractor jobs, monitor their progress, and download and post process the results. Other solvers are either in Beta or are not yet supported for Ansys Cloud.

Uploads to and downloads from Ansys Cloud use archives to improve speed. The Project browser shows both .aedt and .aedtz (archive) files. If you submit an unarchived project, an archive will be automatically created for upload. When the job completes, the results are packaged into an archive. When the download is initiated, the repackaged archive file is transferred to the download folder along with the other job-related files, such as the log.

For single setups, you can use Multi-step submission. This separates the resource specification and reporting as appropriate for each stage of the solve.

Prerequisites for Using Ansys Cloud with Ansys Electronics Desktop

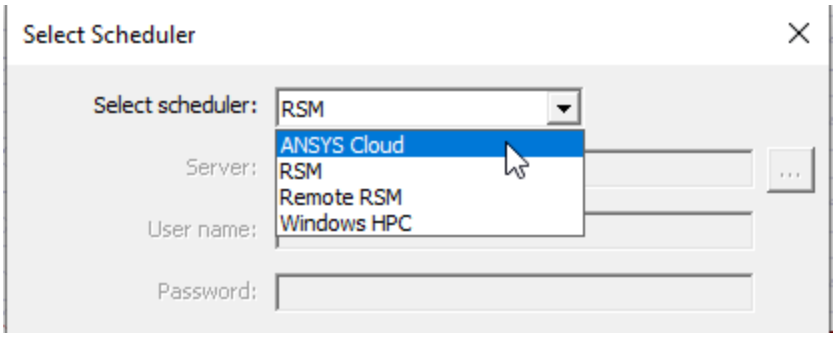
- You will need an "Ansys Cloud Compute Essentials" subscription associated with your Ansys Account. The Ansys Account is an extension of the Ansys Discovery account that includes those products and services that require sign-in, such as the AppStore, Cloud Compute and others. Contact your account manager or service representative to obtain a subscription invitation.
- You must select Ansys Cloud as your Job Management Scheduler, and, if necessary, log in to the Ansys Cloud Service using the Ansys Discovery Account and associated password.

Job Management UI for Ansys Cloud

You can use Ansys Electronics Desktop to submit batch jobs to Ansys Cloud and monitor those jobs.

This involves the following steps:

1. Use **Tools** > **Job Management** > **Select Scheduler** to select Ansys Cloud as the scheduler.



2. Click **Log in** to launch a login window for the Ansys Cloud.

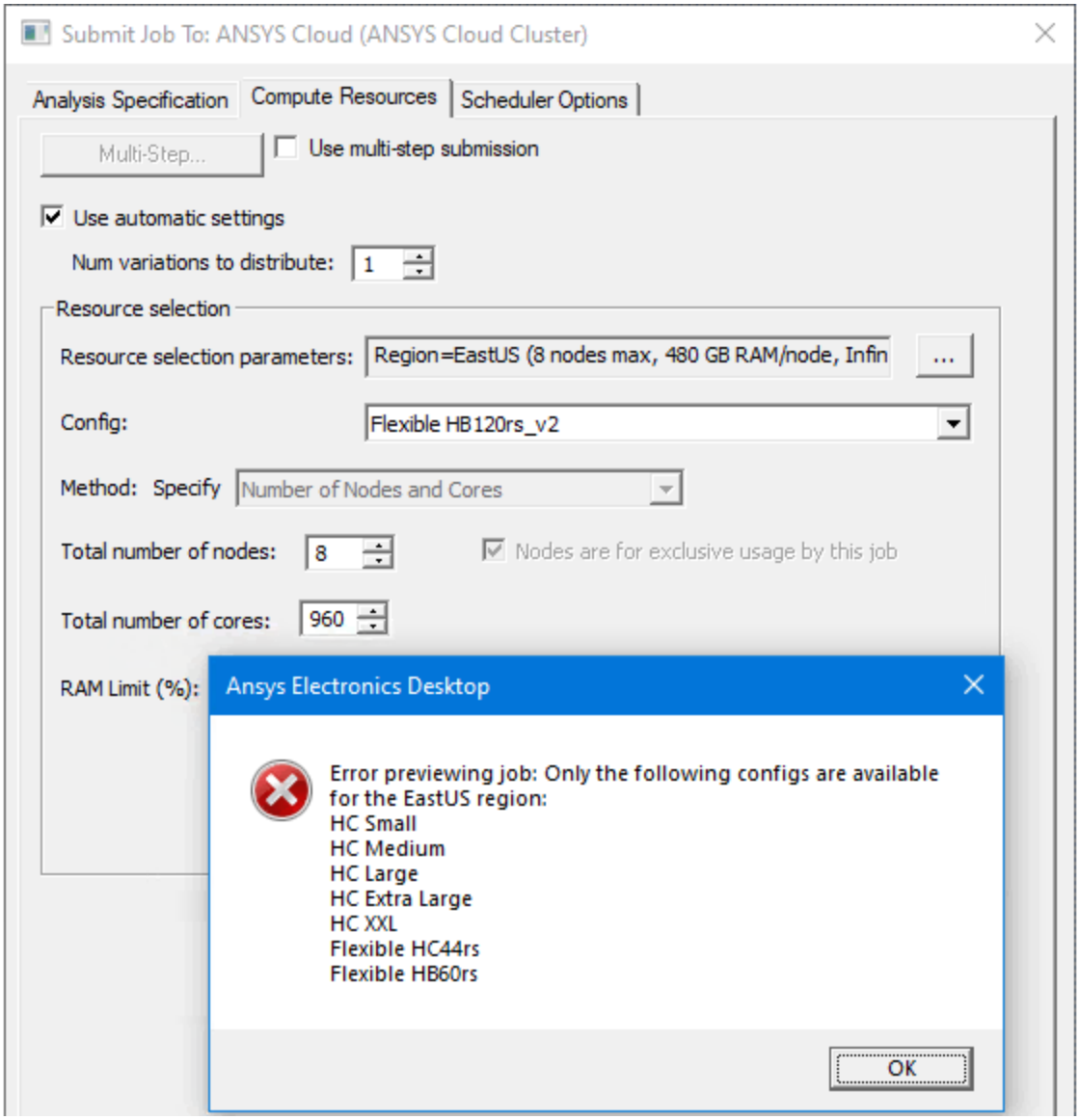
   After a successful login, you may not need to log in again for several days if you do not log out.

   If you wish to log out, click **Log out** from the **Select Scheduler** window.
3. Use **Tools** > **Job Management** > **Submit Job** to submit a batch job to Ansys Cloud.

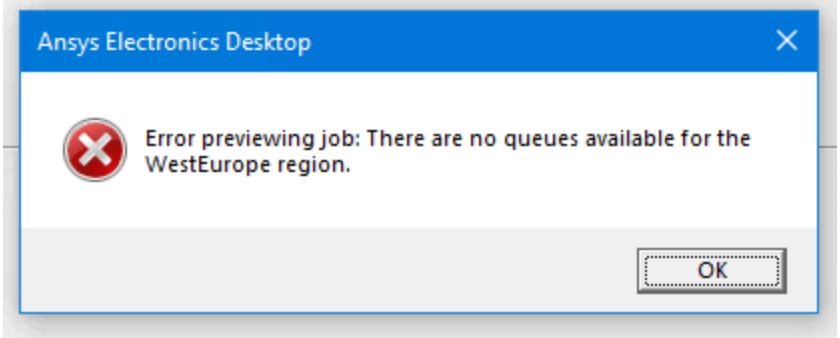
   Ansys Cloud-specific Settings:

   - On the **Compute Resources** tab, **Resource Selection Parameters** will allow you to select a **Region** (for example, EastUS, JapanEast, WestEurope). The **Config** drop-down does not display configurations that are not available in any of the selectable

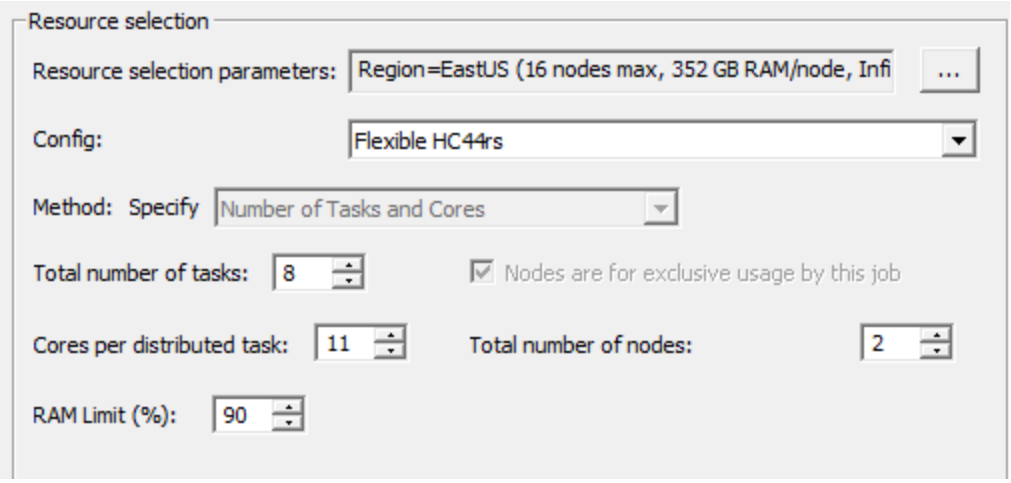
regions. That is, if the regions are EastUS, WestEurope, and SouthwestUS, and none of these regions have the smallhc-based queues, then the "HC Small" configuration does not show up in the **Config** drop-down. If you select a resource that is not available in the specific region you select, an error messages shows the available choices for that region.



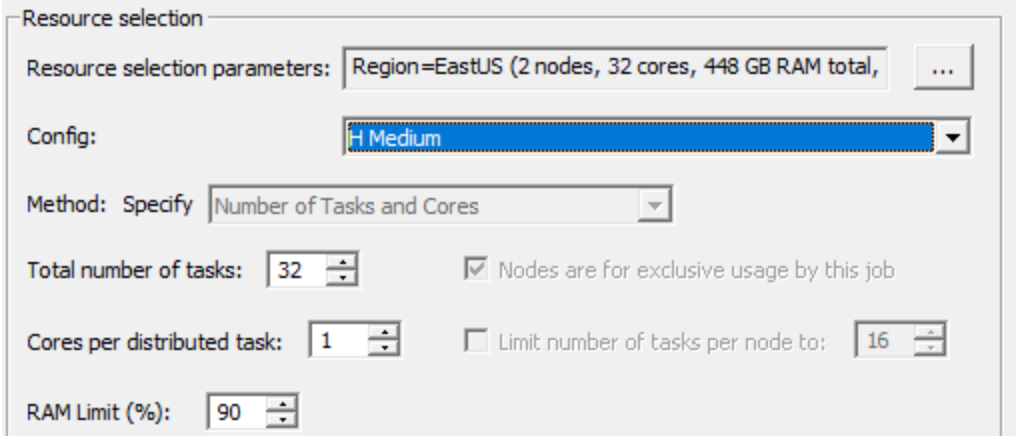
If no queues are available, an error message provides a message:



- On the **Compute Resources** tab, select a **Config** size and style (Flexible or fixed) based on the following:
-
  - **Small:** 16 cores, 1 node, 224 GB RAM total
  - **Medium:** 32 cores, 2 nodes, 448 GB RAM total, Infiniband
  - **Large:** 128 cores, 8 nodes, 1792 GB RAM total, Infiniband
  - **Extra Large:** 256 cores, 16 nodes, 3584 GB RAM total, Infiniband
  - **XXL:** Even larger.
  - **Flexible**: Flexible queues permit the following resource selection methods: Automatic Number of Nodes and Cores, Manual Number of Tasks and Cores. If you select a Flexible queue for manual Number of Tasks and Cores, you will see a new item in **Resource Selection** for "Total number of nodes" replacing the "Limit number of tasks per node to" option that appears for fixed queues.

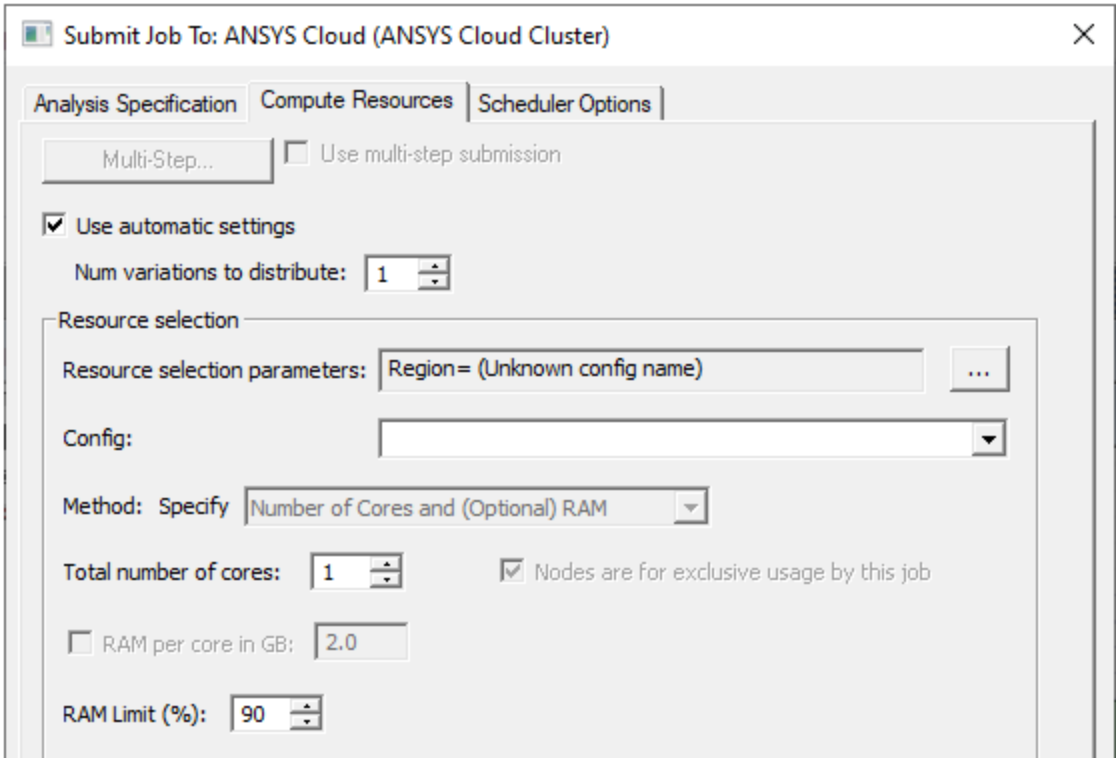


In comparison, if you selected a fixed configuration, the Resource selection panel appeals like this:

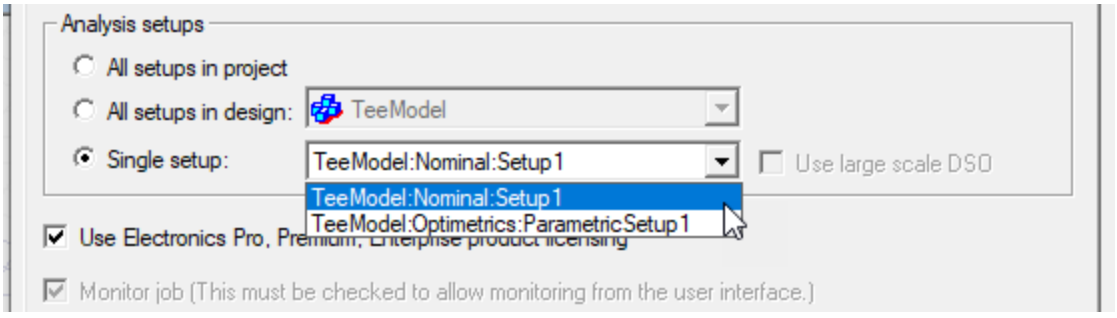


> **Important:**
>
> Attempts to use more for submission will result in error messages.

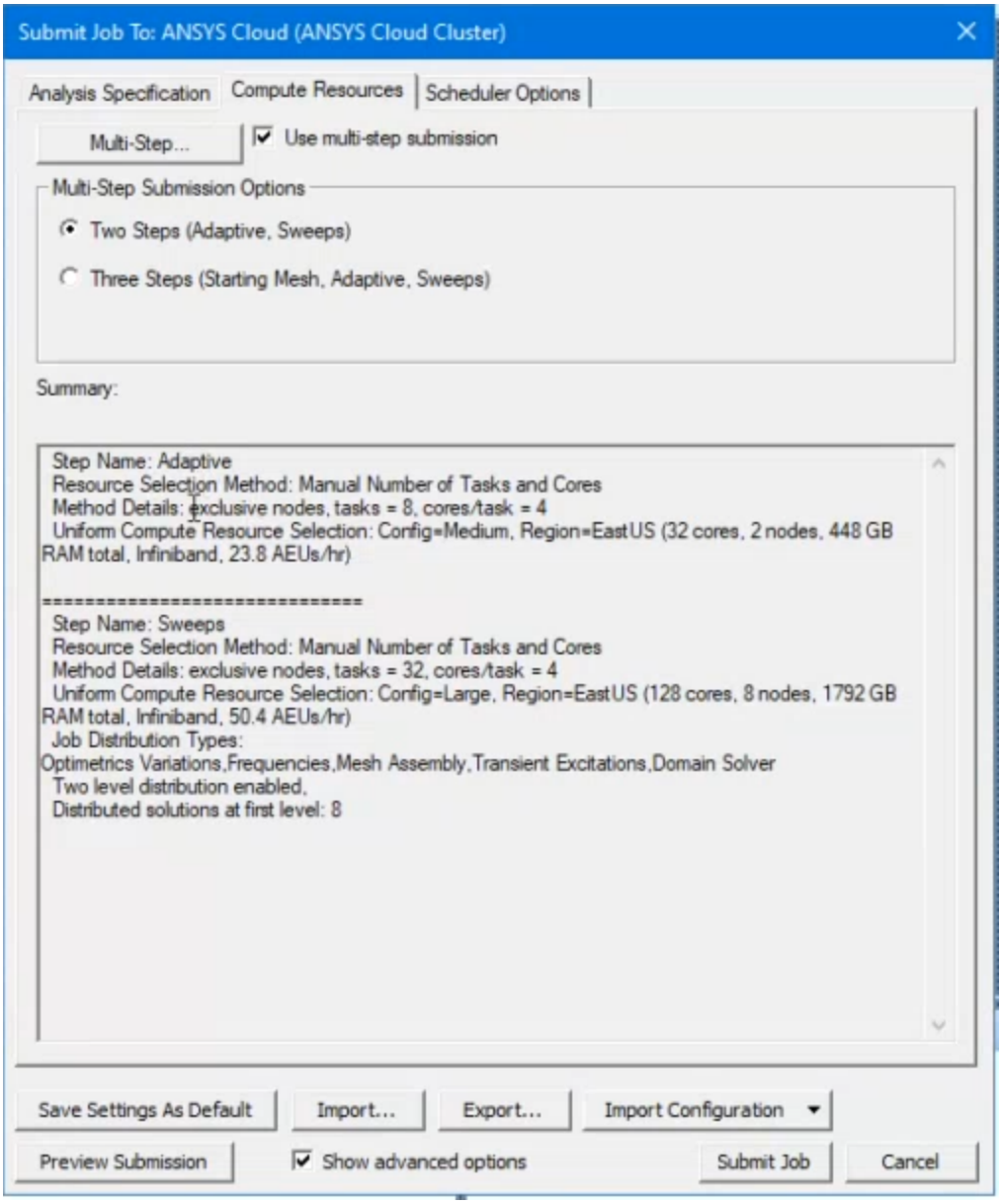
- On the **Compute Resources** tab, if you select **Use automatic settings** with **Num variations to distribute** set to 1, Optimetrics variations will be solved sequentially. Other distribution types will be distributed automatically. If you set **Num variations to distribute** to 2 or more, Optimetrics variations will be solved in parallel. Other distribution types will be distributed automatically.



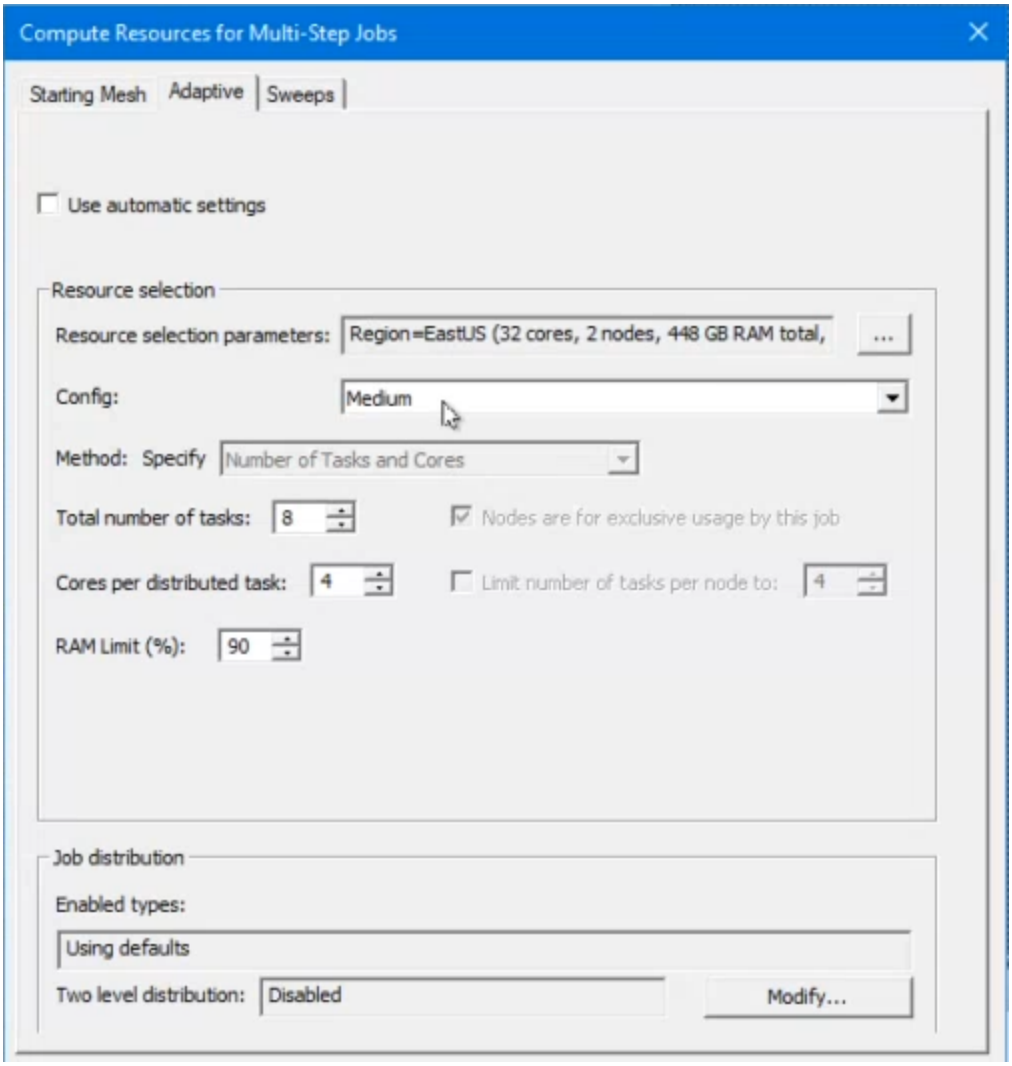
- In response to a set of minimal constraints, Ansys Cloud scheduler may increase the resources assigned beyond the minimal values in order to meet the full set of requirements. For example, if you specify 7 distributed engines with two processors per engine and also limit the number of engines per node to 4, the scheduler may increase the number of cores used in order to meet the limit specified for engines per node. Click **Preview Submission** to see the number of resources assigned, and that the scheduler-generated code includes an MPI specification.
- To use **Multi-Step Submission** on Ansys Cloud, you must specify a single setup on the **Analysis Specification** tab.



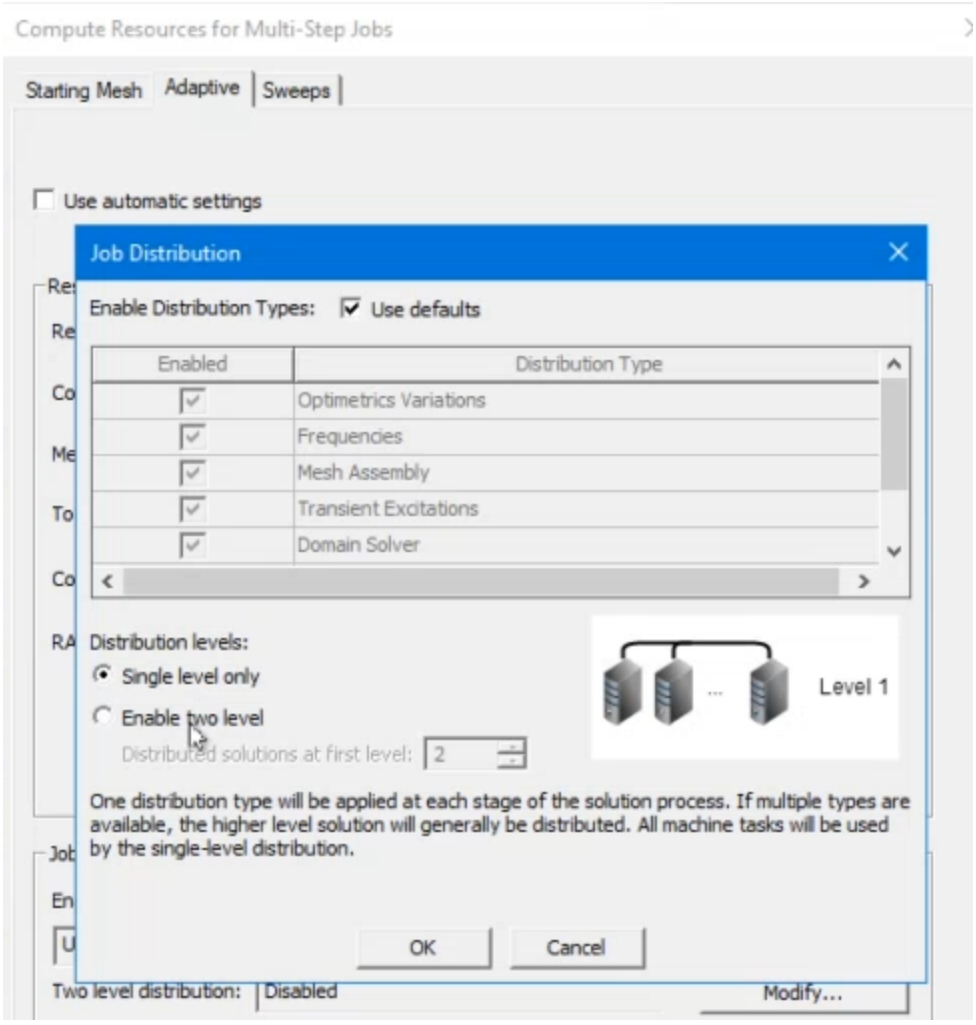
This enables **Multi-Step** options on the **Compute Resources** tab.



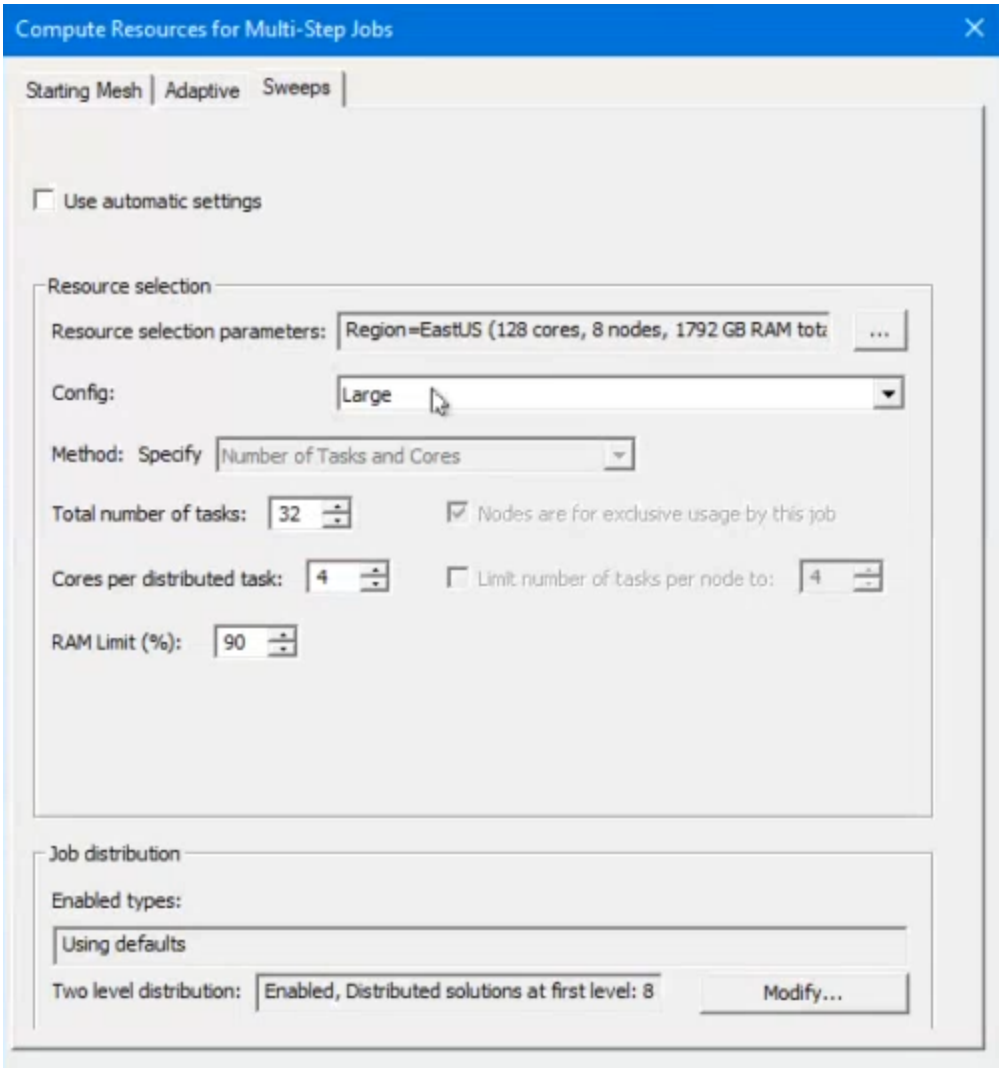
Click the **Multi-Step** button to open the **Compute Resources for Mult-Step Jobs** dialog box. It has tabs for **Starting Mesh**, **Adaptive**, and **Sweeps**.
On the **Adaptive** tab, you can **Use automatic settings**, or specify **Resource Selection**



Click the **Modify...** button on the **Adaptive** tab to open the **Job Distribution** dialog box.

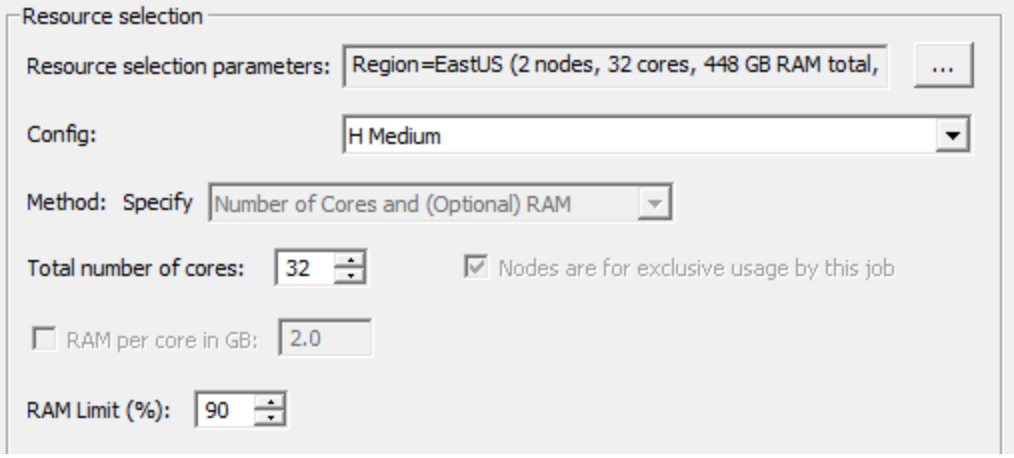


The **Sweeps** tab lets you specify resources or **Use automatic settings**.

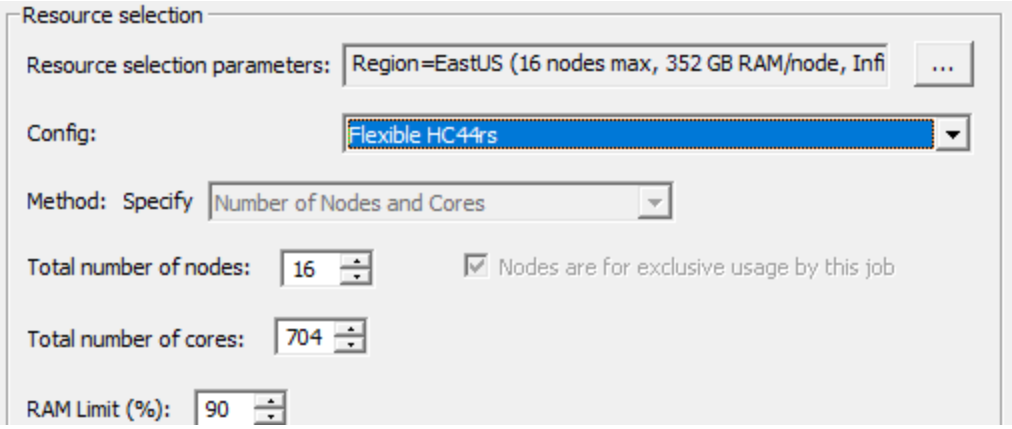


The **Sweeps** tab includes an independent **Modify..** button to specify **Job Distribution**.

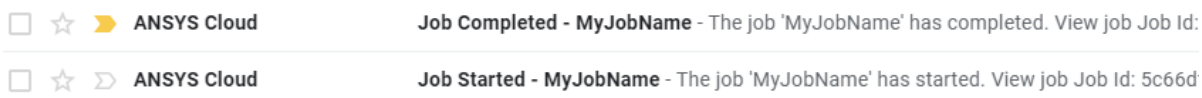
Select automatic settings, or specify different resources for each step. When a job is finished, that is, on the last step, reports update and traces are extracted into CSV files. If select fixed queue for automatic sections, the Resource selection panel only allows cores and has RAM with RAM disabled:



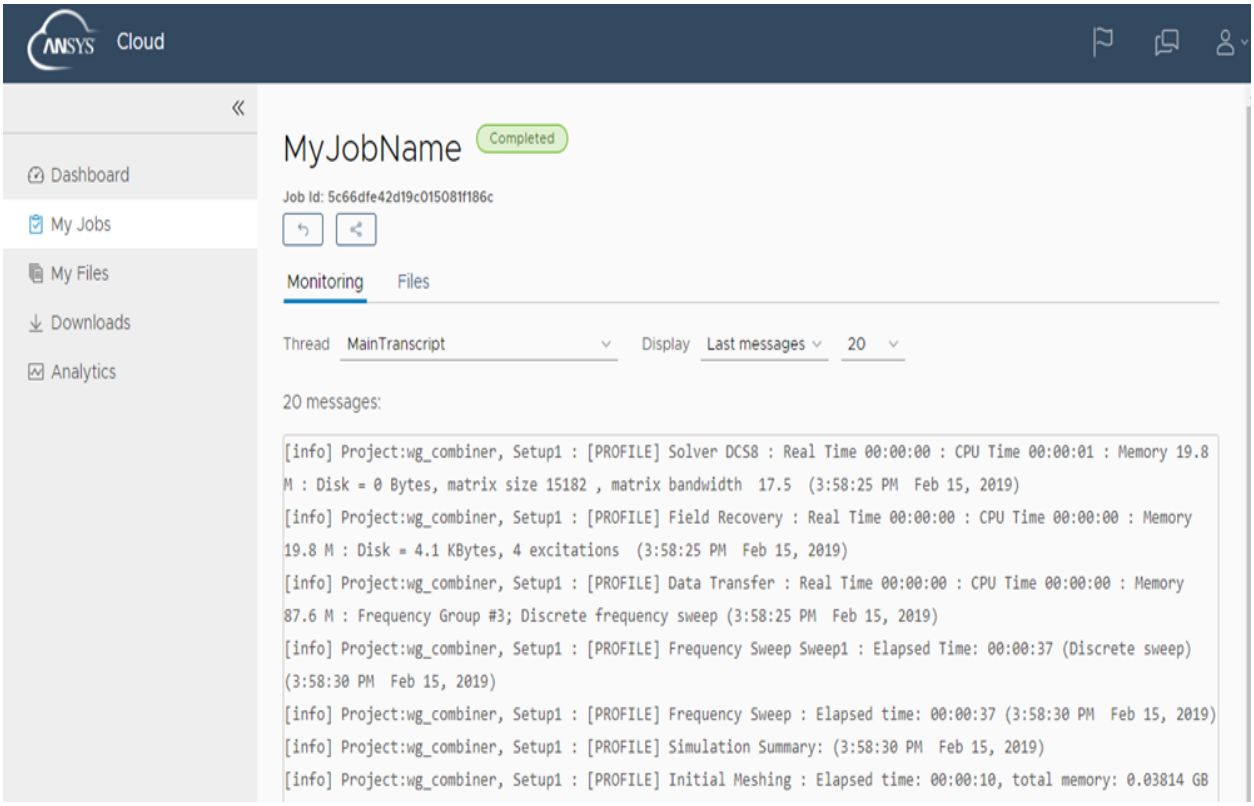
If you select flexible queue for automatic settings, The Resource selection panel only allows nodes and cores for a flexible queue:



4. Use **Tools** > **Job Management** > **Monitor Job** to monitor the job's progress.
5. You will receive an email when the job has started, and another when it has finished.



6. Use the link in the email (or the **Portal** button on the job monitoring window) to launch the Ansys Cloud Portal and view detailed results:



Environment Variable Settings for Ansys Cloud

When submitting jobs to Ansys Cloud from Electronics Desktop, certain environment variable and batchoption settings must be made to ensure proper operation on the cloud. These settings are done automatically through the job submission window.

These environment variable settings will override any attempts from the user to set the same environment variables:

- ANS_NODEPCHECK=1
- ANSYSEM_FIX_REVERSE_LOOKUP=1
- ANSYSEM_DESKTOP_SUBNET_FROM_COMENGINE_ADDR=1

The environment variable ANSYSEM_ENV_VARS_TO_PASS contains a semicolon-separated list of filters for environment variables to be passed from ansysedt.exe to COM engines. You are allowed to set this environment variable. After being prepared for the job it will have the following filters set at minimum (with the user-requested filters following these):

```
ANSOFT_*;ANS_*;ANSYSEM_*;ANSYSLMD_LICENSE_*;I_MPI*
```

Troubleshooting and Debugging Environment Variables

To keep downloaded files that help with troubleshooting, enable the environment variable ANSYS_EM_PRESERVE_DOWNLOAD_FILES.

Debug logs (and other files generated in the working directory) from remote nodes are automatically collected.

Option Settings for Ansys Cloud

The following option is automatically added for Ansys Cloud jobs:

- -autoextract
- For single step jobs it will include "-autoextract reports".
- For multi-step jobs, the last step will include "-autoextract reports" but the first steps will only include "-autoextract" (no reports). Reports are only extracted on the last step.

See: Running Electronics Desktop from a Command Line.

Automatic Batchoption Settings

The license type batch option ("HPCLicenseType") is forced to be "Pack" because Cloud is configured to only work with pack licensing. The following MPI batchoptions are automatically set as "Intel" because Ansys Cloud is configured to only work with Intel MPI.

- HFSS 3D Layout Design/MPIVendor
- HFSS/MPIVendor
- Icepak/MPIVendor
- Maxwell 2D/MPIVendor
- Maxwell 3D/MPIVendor
- Q3D Extractor/MPIVendor

RSM Integration with Job Management UI

Ansys Electromagnetics supports its own Remote Simulation Management (RSM) software, along with other High Performance Computing (HPC) software management programs (see High Performance Computing (HPC) Integration).

When do you need RSM?

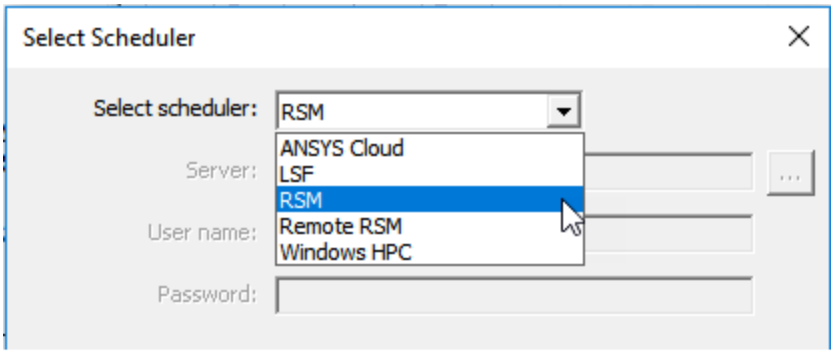
RSM is in general required if you want to run remote or distributed simulations. However, if you have a separate scheduling system that Ansys Electromagnetics supports, and you plan to run batchsolve simulations only, then you may not need to install RSM. For details of installation and configuration of RSM, see the *Ansys Electromagnetics Installation Guides*.

Job Management UI for RSM

You can use the Job Management UI to submit batch jobs to RSM. The Job Management UI is accessed by running Ansys Electromagnetics product Desktop on the designated 'Postprocessing node' of the cluster. The Desktop provides UI commands for Scheduler selection, Job submission and Job monitoring/control. You access the Scheduler User Interface by clicking **Tools>Job Management>Select Scheduler...**

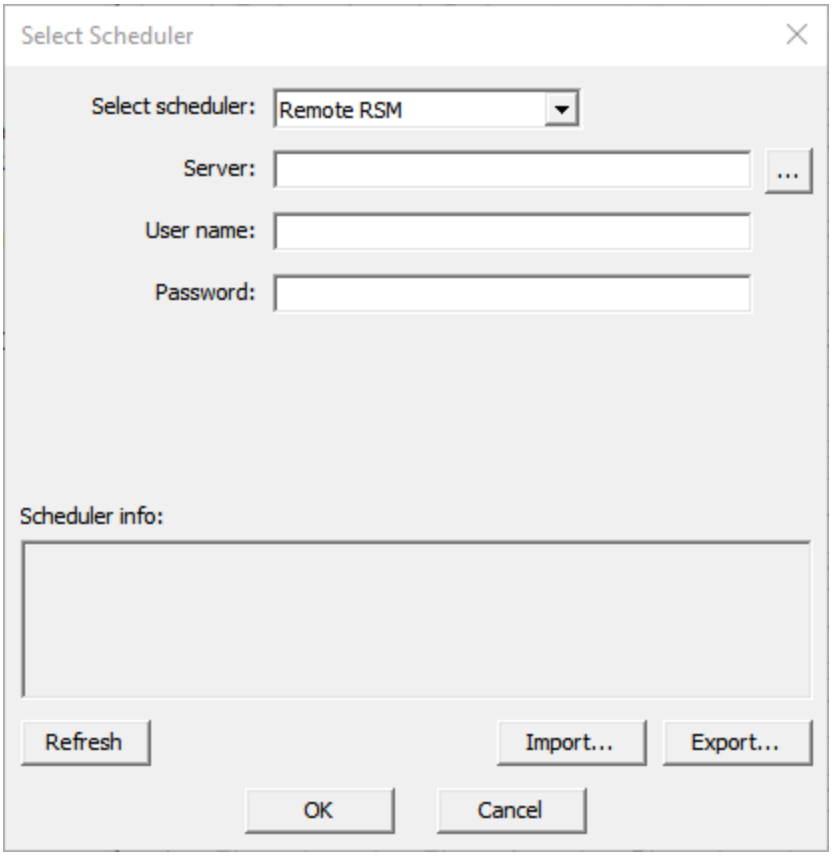
You can also select the **Simulation** tab of the ribbon, and click the **Scheduler** icon.

This displays the selection dialog box. The **Select scheduler** drop-down menu lists potential schedulers (which can include RSM, LSF, Windows HPC, or sge, depending on the environment).

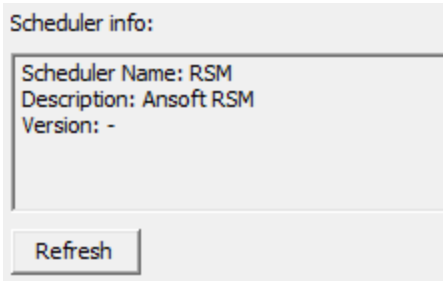


If you select a scheduler that is not supported in your environment, you receive a warning message.

If you select Remote RSM and your environment has been configured , you can select a computer, user name, and password.



After selecting a scheduler, you can click **Refresh** to display information for that scheduler.



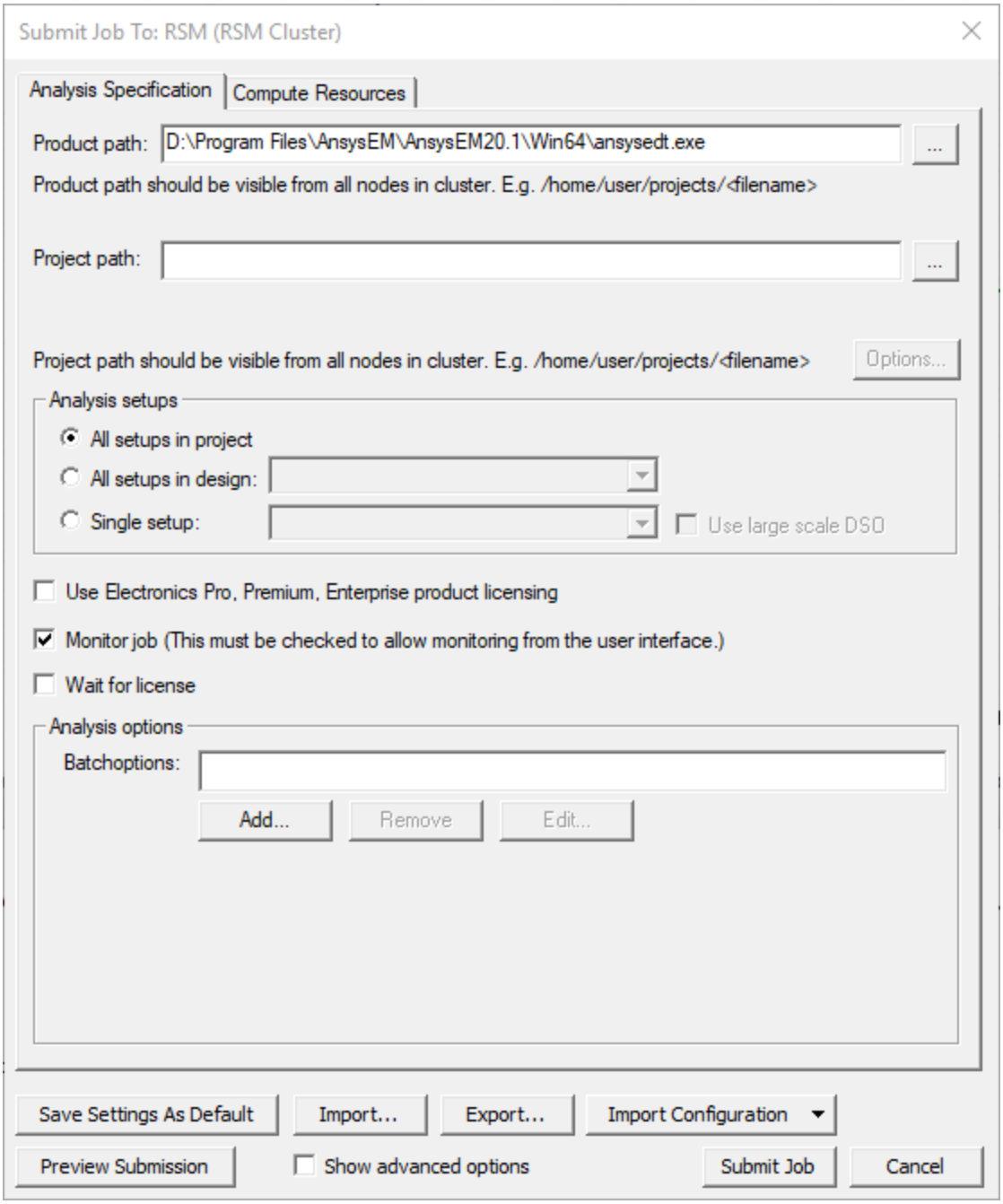
Once you have selected a scheduler supported in your environment, you can go through the following steps to submit a batch job.

1. Setup and prepare the model on local workstation.
2. Copy the input project (or folder, if the project references external files) from a personal workstation to a shared-drive on cluster (for example, project is copied to /home/projects/spool/test.aedt).
   - In the RSM environment, you are required to specify a machine-list. (See the Remote Analysis options panel.) For example, say the machine-list is: 3 cores from 'm1' and 3

cores from 'm2', for a total of 6 engines. You select the list on the **Compute Resources** tab of the **Submit Job to RSM** dialog box, as described below.

3. Open a remote-desktop session (or equivalent such as vnc session) on the node corresponding to the first machine of job's machine-list, 'm1' in this case. Launch Desktop graphically on 'm1'.
4. Select **Tools>Job Management>Submit Job...** or **Project> Submit Job...** or [ProductName]> **Submit Job...** to open the **Submit Job To** dialog box. You can also access **Submit Job** from the shortcut menus for the Project Name, Design name, Analysis

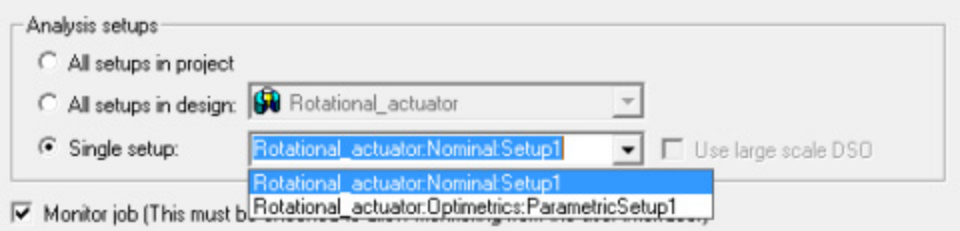
Setup, or Optimetrics Setup.



- The **Analysis Specification** tab has parameters to specify the input project model, the analysis setup and analysis options (including batchoptions) that affect analysis algorithms.
- The **Compute Resources** tab specifies the amount of compute resources and how to select specific resources from the available pool.

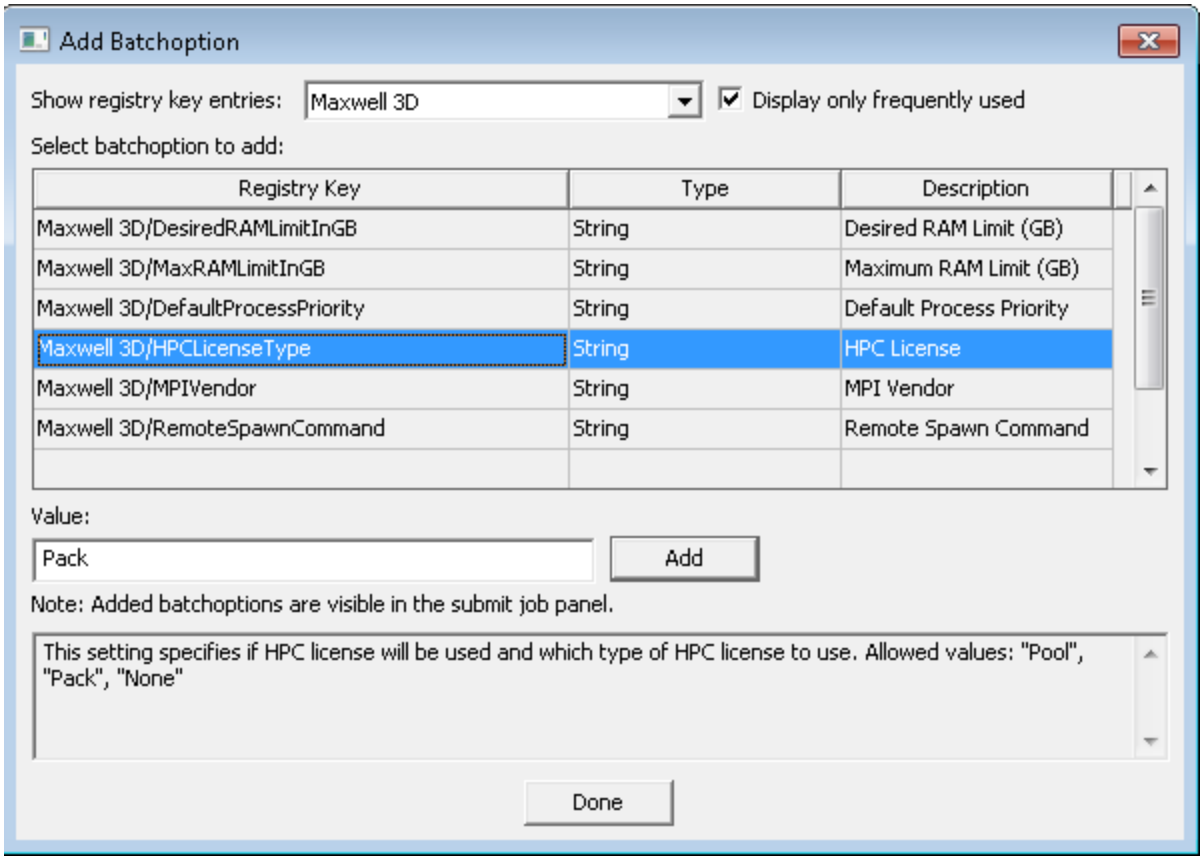
5. Use the ellipsis button [...] to open a browser to select the project. The Project can be an archive.

6. In the Analysis setups field, you can select radio buttons for All setups in the project, All setups in the design, or a Single setup. For instance, the Rotational_actuator.aedt example includes setups for Nominal, Parametric, and Optimization.

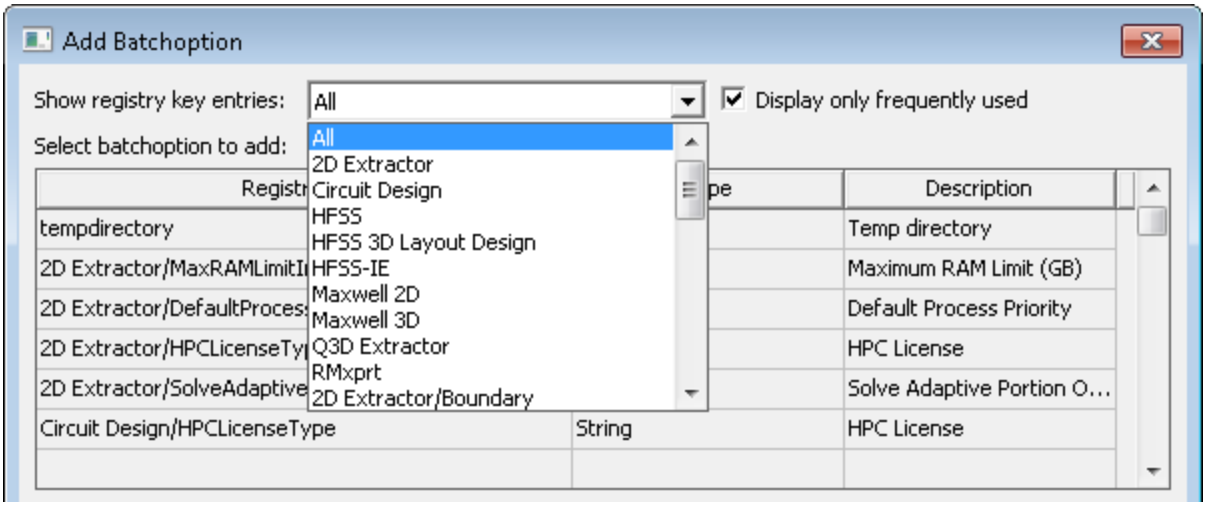


   For Parametric setups, you have the option to select Use Large Scale DSO. For details on how and when you use this feature, see Job Management Interface for Large Scale DSO.

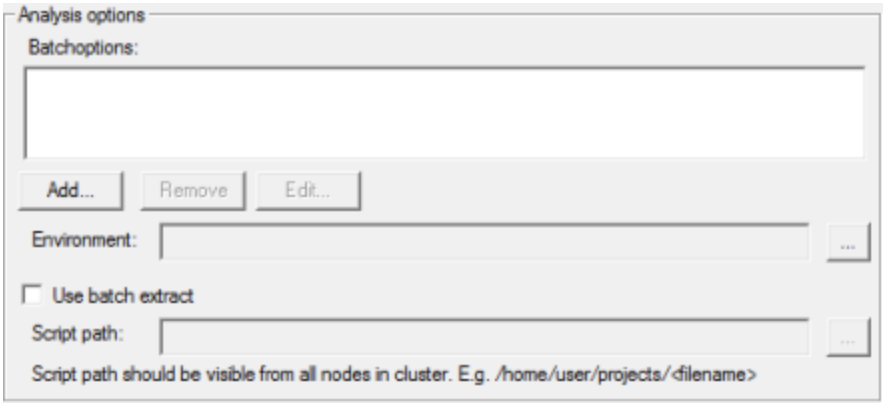
7. The Analysis options includes checkboxes for Monitoring the job, whether to wait for a license, and a field for adding Batchoptions. via a graphical interface, or as text.
   - If you intend to Monitor the job through a user interface, you must check Monitor job. You can then monitor this job through the **Tools>Job Management>Monitor Jobs...** command or by checking the dialog that opens when you submit the job.
   - The Batchoptions field allows you to add additional -batchoptions parameters, either as text, or by using a dialog with selection menus. Click the **Add** button to view the **Add Batchoption** dialog box.



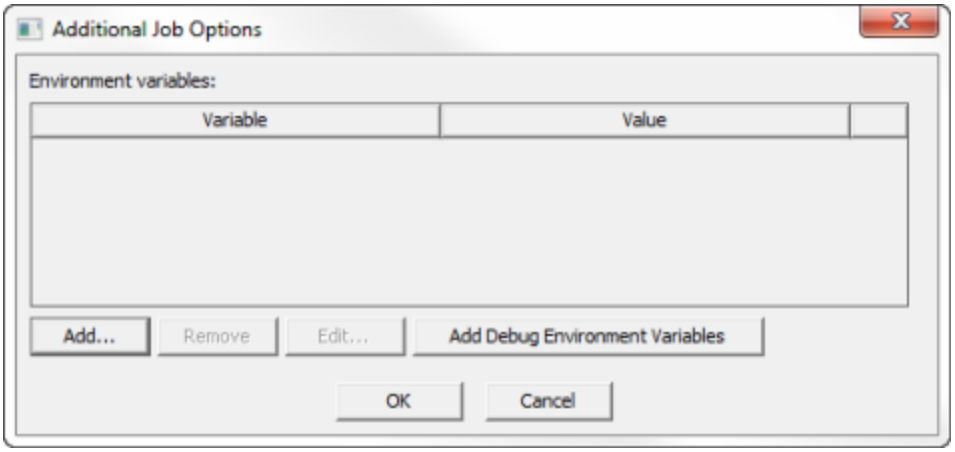
The Show registry key entries field lets you filter the entries displayed, by means of drop down menu selection, and a check box to Display only frequently used entries.



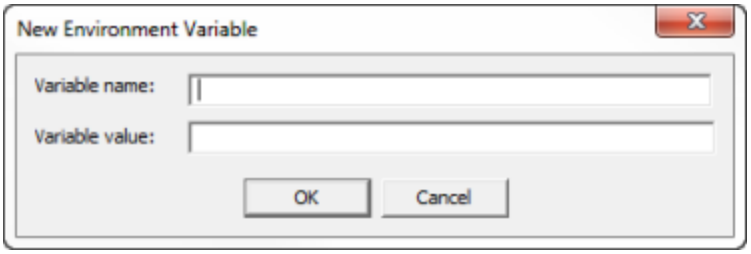
- When you have selected a batchoption, you can type the value in the field, and click the **Add** button to add the option to the batchcommand.
- In the **Submit Job To:** dialog box, you can enable Show advanced options to display additional fields for Environment variables, and whether to Use batch extract.



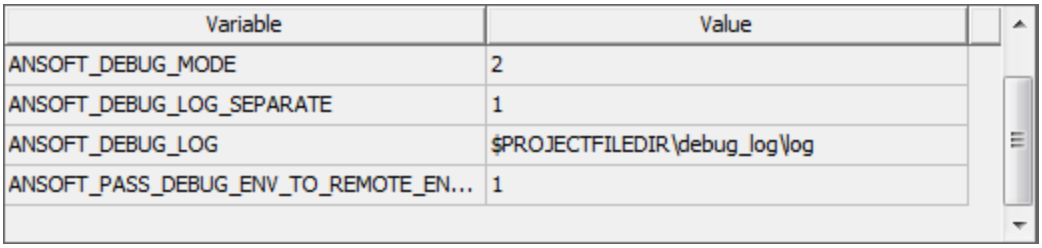
This field is for environment variables, for instance, for debugging features or other variable controlled features. Click the ellipsis [...] button to open a dialog for **Additional Job Options**.



The **Add...** button opens a New Environment Variable dialog in which you can include a variable name and value.



Clicking the **Add Debug Environment Variables** button automatically adds a set of debug variables. This can be useful in working with Ansys Application Engineering support.



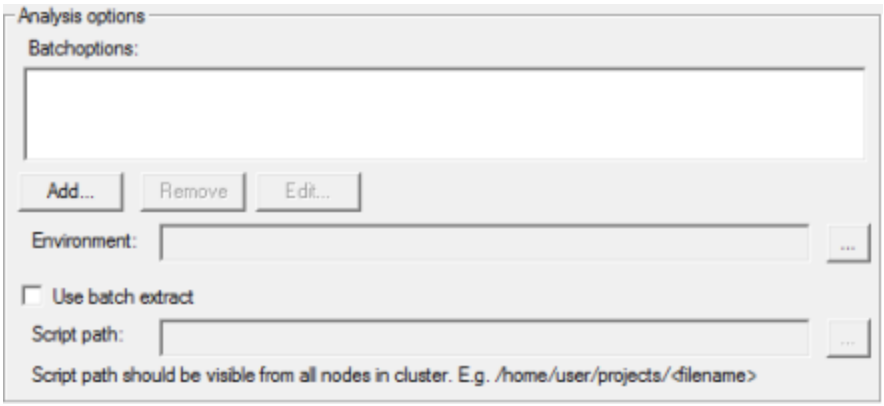
Selecting a variable in the dialog enables the Remove and Edit buttons. The Edit button opens a dialog where you can change the variable and value.

8. To see the command-line to be submitted to the scheduler, click **Preview Submission.** This opens a dialog showing the command to be sent to the scheduler.

The text can be copied to the clipboard, if desired.

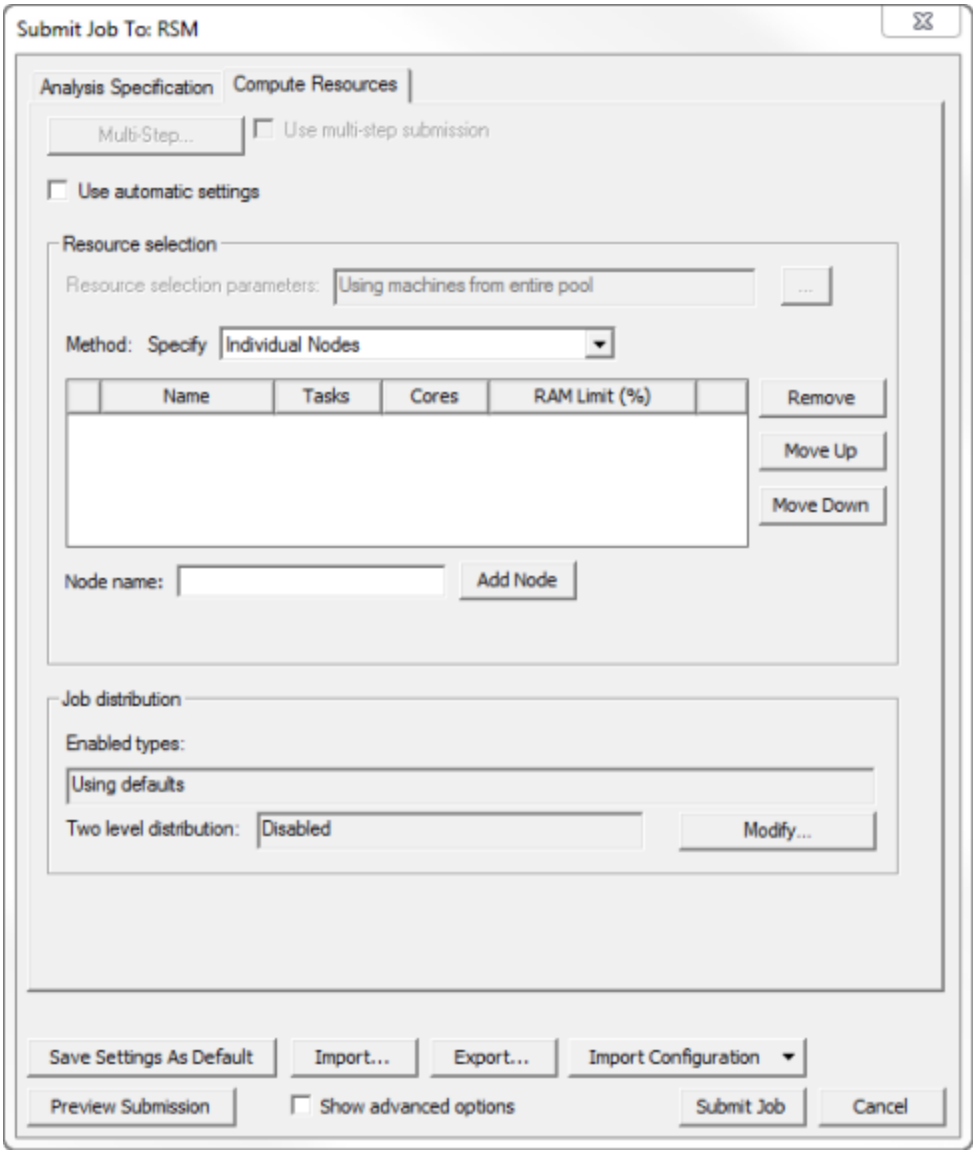
Use Batch Extract for RSM

Selecting Show advanced options for RSM also shows the Use batch extract fields.

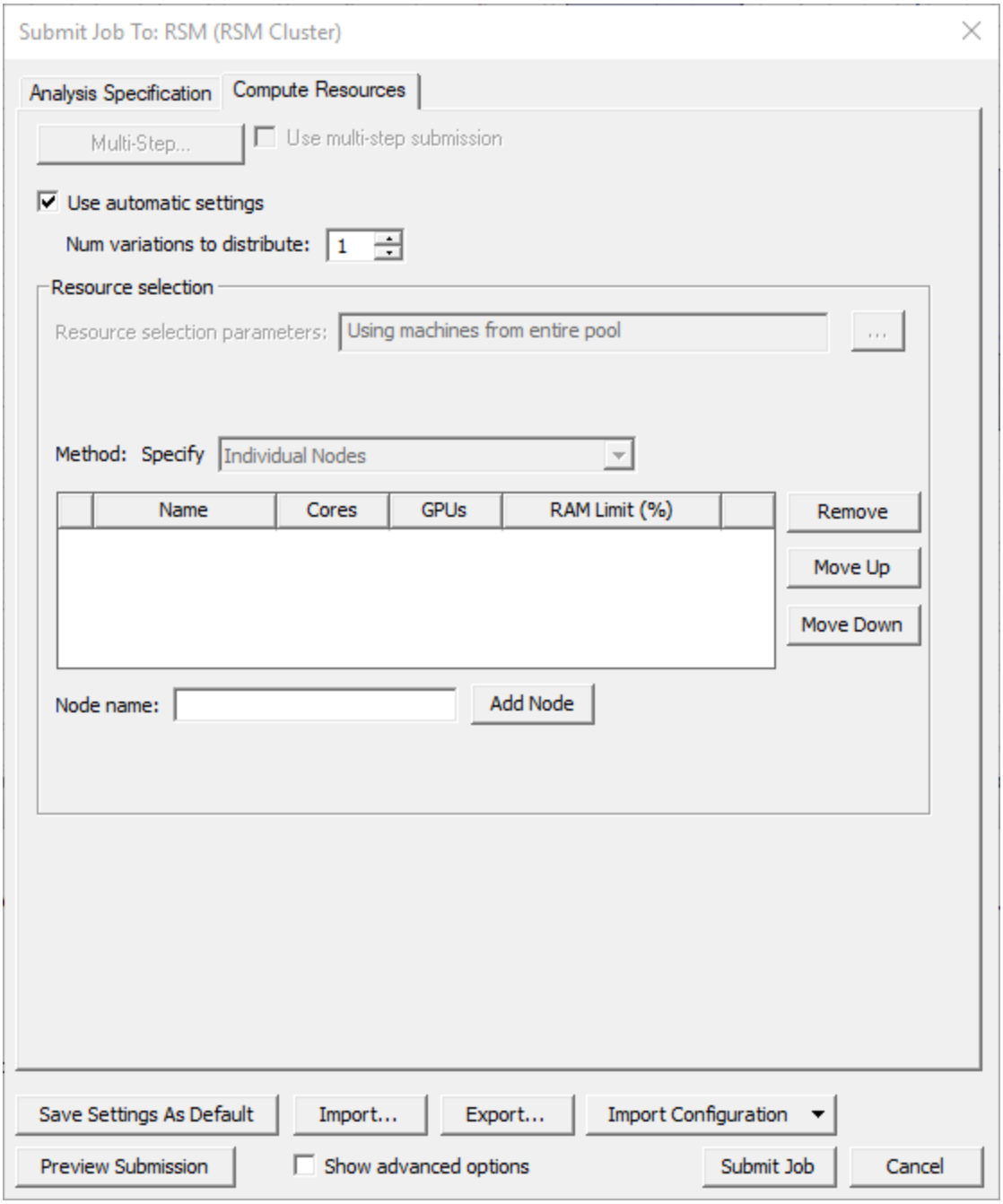


See the discussion on Running Maxwell from a Command line for a discussion of the solve information available through batch extract.

9. The **Compute Resources** tab of the **Submit Job to: RSM** dialog displays other parameters. Depending on the resources available for a scheduler environment, some of the fields may be disabled.



With **Use automatic settings** selected, the Job distribution field is removed and the Use automatic settings checkbox and Num variations to distribute field appears.

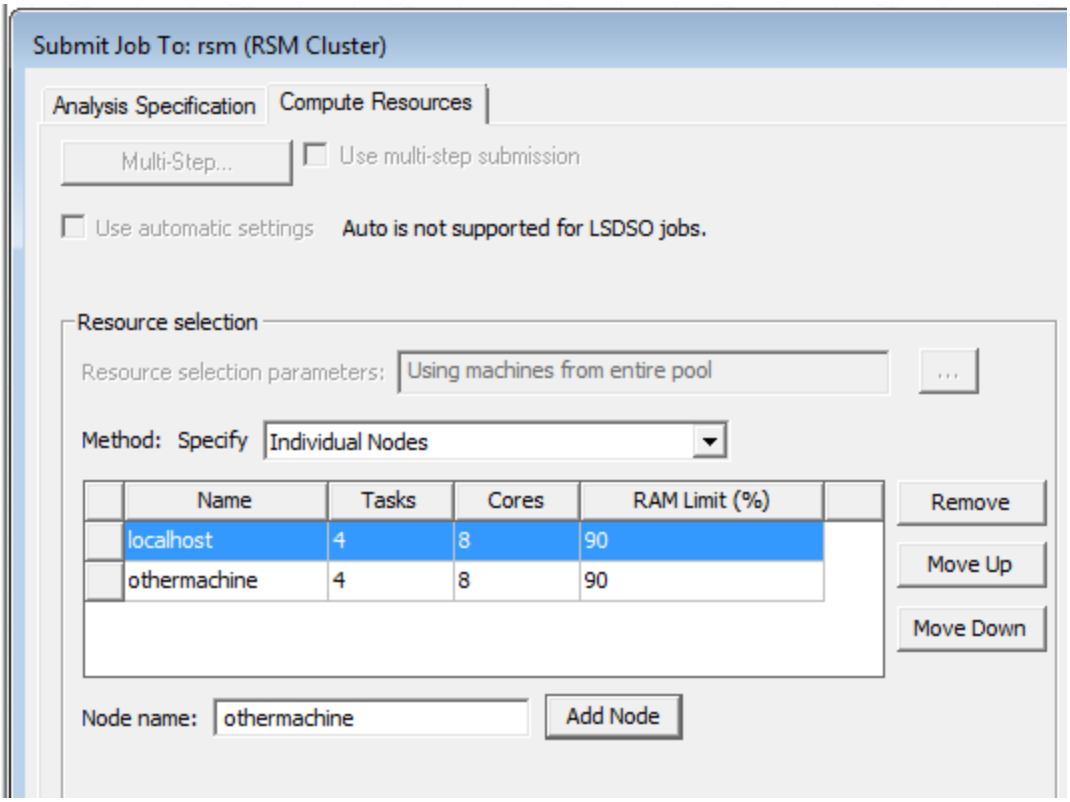


Note	If you select Use automatic settings with Num variations to distribute set to 1, Optimetrics variations will be solved sequentially. Other distribution types will be distributed automatically. It does distribute frequencies, domains, and use of multiple level domains. If you set Num variations to distribute to 2 or more, Optimetrics variations will be solved in parallel. Other distribution types will be distributed automatically.

Otherwise:

- Specify node list

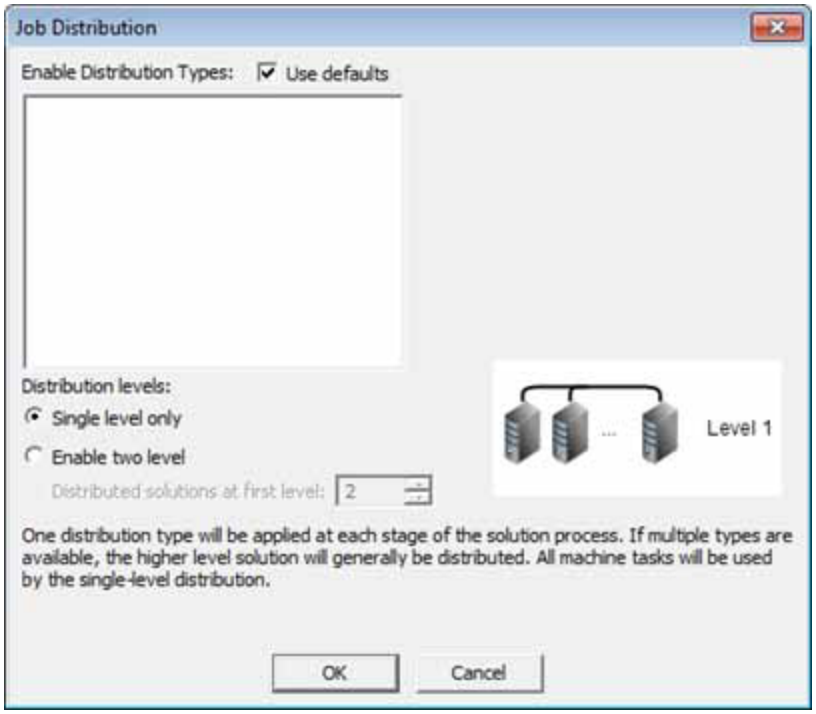
  Here you can specify a node list. In a computing environment where the available cores are not uniform, you can use this to control which resources your job will use. For use with Large Scale DSO for RSM, for jobs that are submitted from job submission panel, localhost must be the first node in the resource selection node list, other wise LSDSO solve with RSM will fail.



Job Distribution

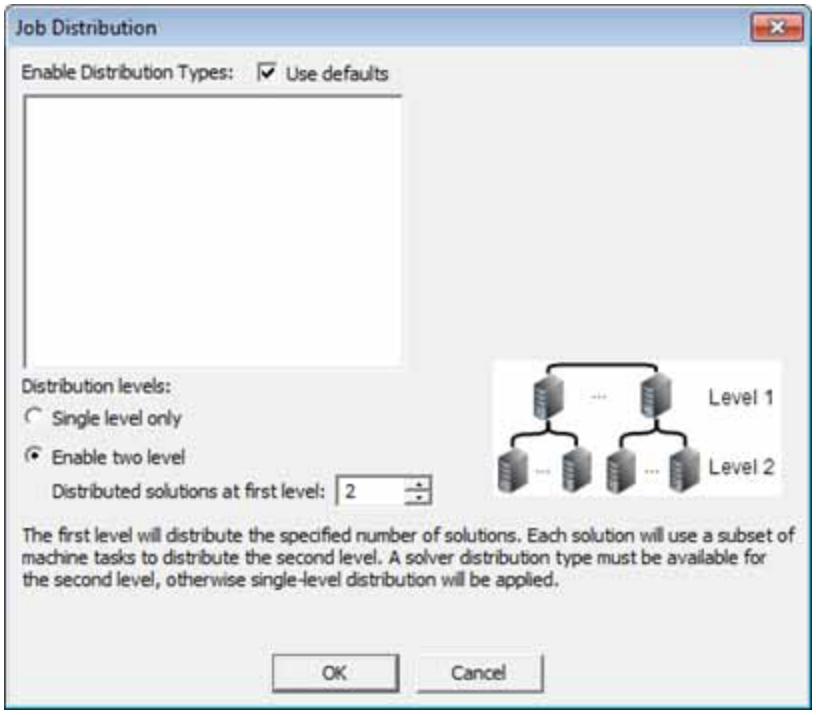
If you disable User automatic selection, you can modify the Job distribution settings.

Two level distribution, which may be disabled. Click the **Modify** button to display the **Job Distribution** dialog.



Enabled Distribution types can be modified here.

Second level distribution operates within DSO. If available and enabled you can specify a number of engines for level 1.

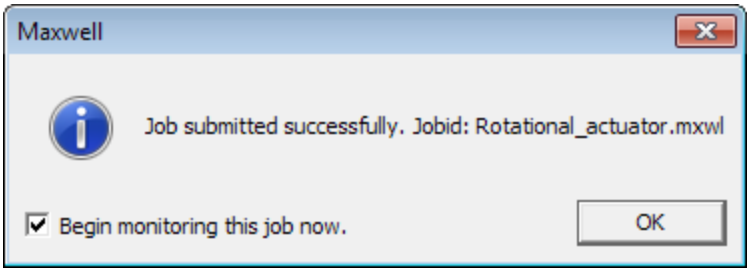


In response to a set of minimal constraints, the Scheduler may increase the resources assigned beyond the minimal values in order to meet the full set of requirements. For example, if you specify 7 distributed engines, with two processors per engine, and also limit the number of engines per node to 4, the scheduler may increase the number of cores used in order to meet the limit specified for engines per node. Notice that a preview of the Submit Job Results shows the number of resources assigned, and that the scheduler generated code includes an MPI specification.

10. To submit the command with the specified parameters, click **Submit Job**.

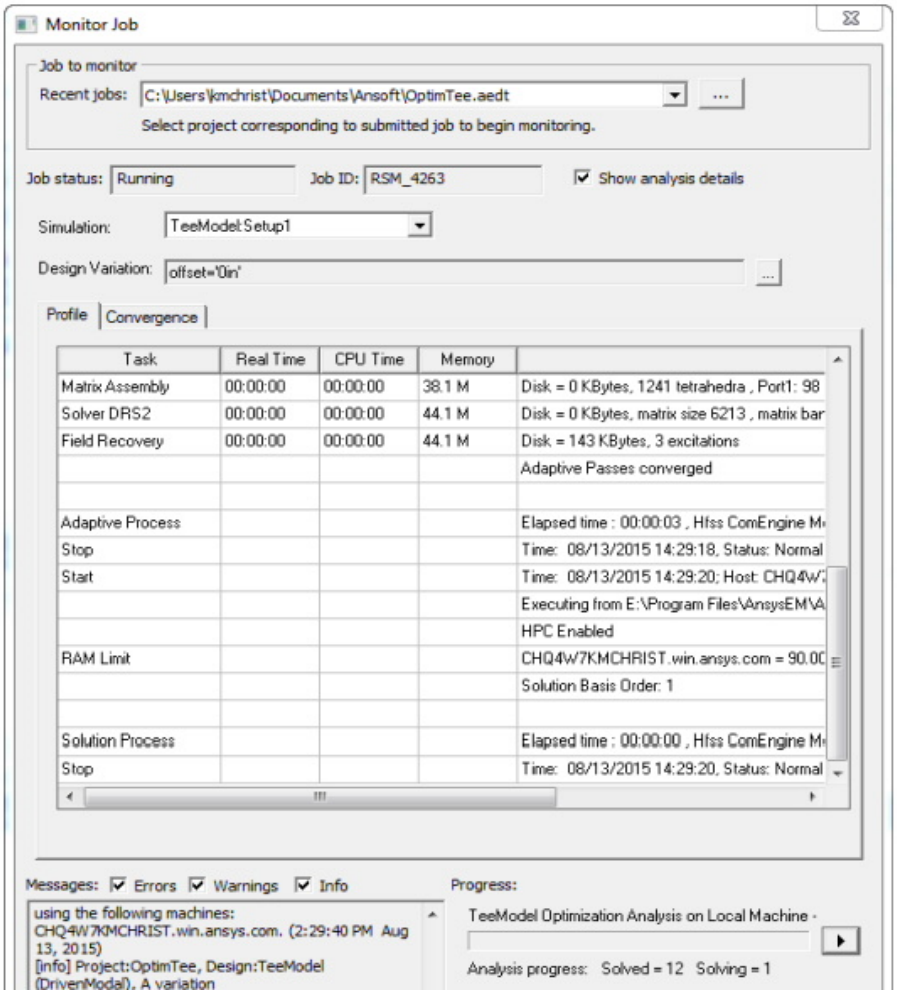
Note	The RSM environment does not support for queuing, so 'Submit Job' will immediately start running the job.

A dialog displays in which you can check "Begin monitoring this job now."



11. You can monitor this job either automatically (by checking the option) or through the **Tools>Job Management>Monitor Jobs...** command. For more details, see Monitor Job

window.

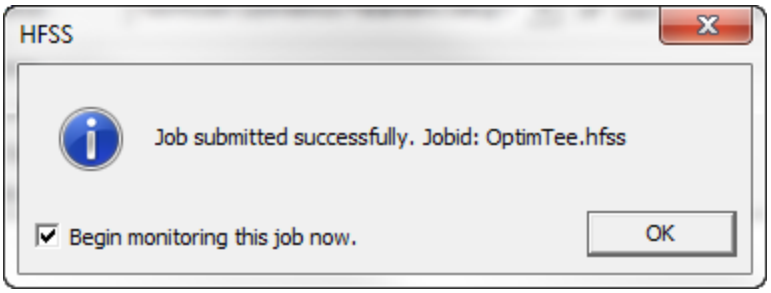


The dialog contains fields reporting the job status, job ID, messages issues, and progress. You can filter the messages for Errors, Warnings, and Info. By option you refresh the job manually or automatically at specified intervals.

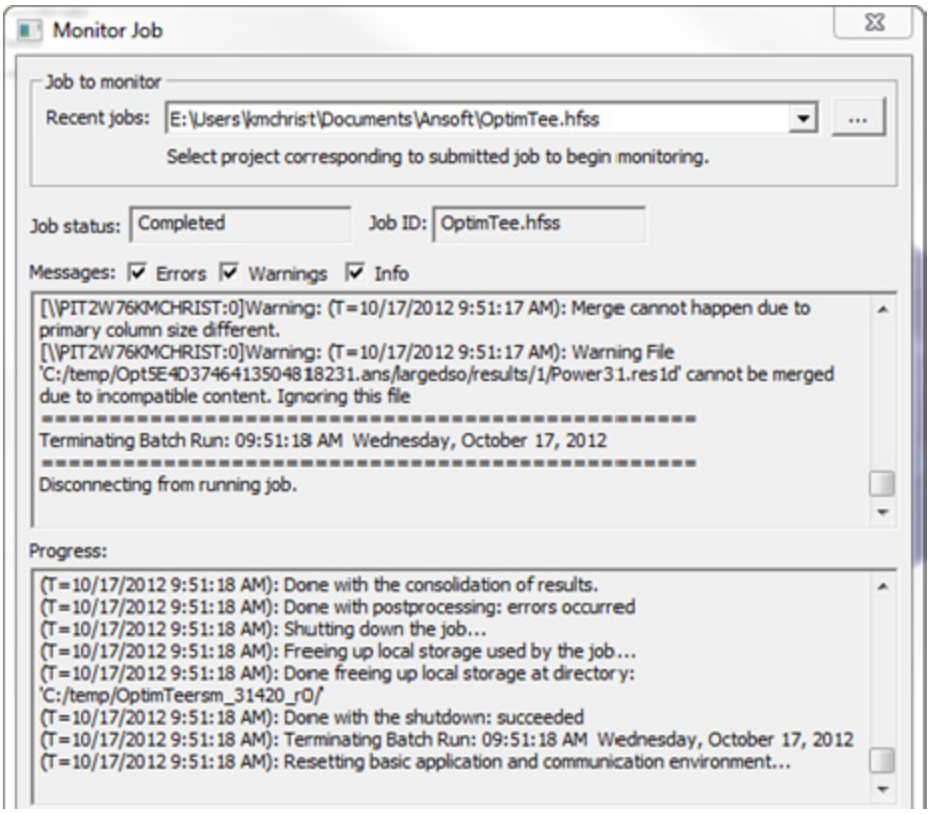
12. To submit the command with the specified parameters, click **Submit Job**.

Note	The RSM environment does not support for queuing, so 'Submit Job' will immediately start running the job.

A dialog displays in which you can check "Begin monitoring this job now."

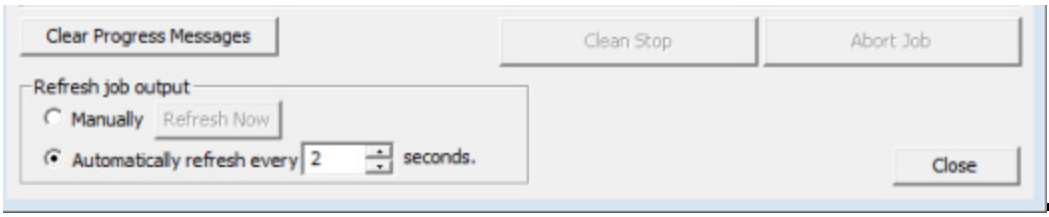


13. You can monitor this job either automatically (by checking the option) or through the **Tools>Job Management>Monitor Jobs...** command.



The dialog contains fields reporting the job status, job ID, messages issues, and progress. You can filter the messages for Errors, Warnings, and Info.

By option you refresh the job manually or automatically at specified intervals. You can choose to **Clear Progress Messages**.



You also have command buttons to perform a **Clean Stop** of a simulation between time steps, or to **Abort** a simulation.

Process for Changing the Listening Port used by AnsoftRSM Service

To change the listening port used by the AnsoftRSMService, you need to change the configuration file, ansoftrsmservice.cfg, as follows:

You must specify the ListenPort within a 'CommDetails' block, which must be within a 'Default:CommDetails' block, which must be within the top level block of the file, the 'AnsoftCOMDaemon' block. The following example shows the listen port changed from 32958 to 32957, with these blocks at the beginning of the file:

```
    $begin 'AnsoftCOMDaemon'
$begin 'Default:CommDetails'
  $begin 'CommDetails'
ListenPort='32957'
  $end 'CommDetails'
$end 'Default:CommDetails'
. . . .
    $end 'AnsoftCOMDaemon'
```

For the second level block, ensure that there is a single colon character and no spaces or tabs separating the two parts of the block name 'Default:CommDetails'. The third level block, with name 'CommDetails' is also required. Use caution when editing this file by hand, because any typos in the block or value names may cause the data to be ignored.

Using the Ansys EM HPC Diagnostics Tool

The Ansys EM HPC diagnostics tool simplifies HPC troubleshooting by automating diagnosis of routine issues. The diagnostics tool is run on the cluster as a scheduler managed job. Using its HTML-based diagnostics report, the cluster administrator or Ansys support staff can either resolve the issue, or guide the user with steps for further troubleshooting. In some cases, Ansys support staff may request to rerun the diagnostics with additional diagnostics tests. The user may extend the diagnostic scripts to suite their HPC environment.

The following sections describe how to use the diagnostics tool.

- Supported schedulers
- Running the diagnostics job
- Basic diagnostic job
- Using diagnostics scripts on Linux clusters
- Using Windows HPC job file
- Diagnostic report
- Site-specific diagnostics job
- Environment variables
- ANSYSEM_DIAG_PROD_DIR contents

- ANSYSEM_DIAG_RESULTS_DIR contents
- How does the diagnostic tool work

Supported schedulers

The tool supports diagnosis of issues on Linux and windows clusters managed by the following schedulers:

- LSF
- SGE
- PBS/Torque
- Windows HPC

For the above schedulers, the tool includes standard diagnostic scripts. Further, if password-ssh has been enabled, it also supports generic Linux clusters using ssh. Please note that currently diagnostics tool does not support PBSPro and LSF/Windows.

Running the diagnostics job

The diagnostics are run as a scheduler managed job. Once the job finishes, you locate the resulting HTML file and provide it to the cluster administrator or to Ansys support staff. If there are any job or test failures, please also provide the networking*.json files from the Hosts subdirectory as well.

Basic diagnostic job

To run the basic diagnostics, submit a diagnostic job to the scheduler using a provided job submission script. Each standard diagnostic job is a 12 core job with 4 cores per host. On Linux, running this script submits a scheduler job to run the diagnostic tool on the cluster. On Windows, you need to submit a job using a job file.

Basic scripts for each supported scheduler are available in the diagnostics subdirectory of the schedulers directory.

Linux:

```
.../Linux64/schedulers/diagnostics
```

Windows:

```
.../Win64/schedulers/diagnostics
```

Using diagnostics scripts on Linux clusters

The following standard scripts are provided in the diagnostics directory:

```
.../Linux64/schedulers/diagnostics.
```

These job submission scripts are scheduler specific.

Scheduler	Standard job submission script	Comment
LSF	test_lsf	Supports both lsrun and blaunch
SGE	test_sge	Supports both qrsh and rsh
PBS/Torque	test_torque	Requires changing the PATH and PBS_BINARY_PATH environment variable
Generic Linux cluster	test_ssh	Supports only ssh. Requires password-less ssh. Requires creating a file with the names of hosts and saving it in ${HOME}/ansysem_hostfile

Using Windows HPC job file

A sample job file **winhpctest.xml** is available in the diagnostics directory:

```
.../Win64/schedulers/diagnostics
```

To submit this diagnostic job, you must change the job description to suite your environment as following:

1. Select a directory for saving the diagnostic results. This directory must be accessible at the same path from all the hosts of the cluster.
2. Locate the directory for Ansys EM installation. This directory also must be accessible at the same path from all the hosts of the cluster.
3. Locate the **winhpctest.xml** in the diagnostics subdirectory of schedulers directory in the Ansys EM installation.
4. Start Windows HPC job manager, and choose “New job from XML File…” action.
5. Select the **winhpctest.xml** job file.
6. Change the value of both the following environment variables with the directories located in the first two steps:

   ANSYSEM_DIAG_PROD_DIR

   ANSYSEM_DIAG_RESULTS_DIR
7. Now submit the job.

Note	After making the above changes, you can also save the resulting XML file using "Submit Job XML File…". Then you can submit the job using the job command as following: `job submit /jobfile: XMLfileName`

Diagnostic report

The diagnostic report is an HTML file which (along with other related diagnostics results) is placed in the following directory

Linux:

```
${HOME}/Ansoft/HPCDiag/Results/JOBID
```

Windows:

```
%ANSYSEM_DIAG_DIR%\Results\JOBID
```

Report file:

```
.../HTML/report.html
```

where JOBID is the job ID assigned by the scheduler. On Windows, the user must specify ANSSEM_DIAG_DIR directory.

Site-specific diagnostics job

To run a diagnostic job with job submission parameters of your choice, you need to create your own job submission script. For example, you may want to specify a different LSF queue, or select a different SGE parallel environment. To run such a job, you need to create your own job submission script starting from the standard diagnostic scripts with the following steps:

1. Locate the relevant standard diagnostic script in the diagnostics subdirectory of schedulers directory in Ansys EM installation.
2. Make a copy of the diagnostics script into a directory that is accessible from a submit host for the cluster.
3. Edit the script file to change the value of ANSYSEM_DIAG_PROD_DIR environment variable to point it to the installation directory (See below).
4. Modify the job submission parameters as needed.
5. Optionally, copy any site-specific diagnostic tests provided by Ansys support staff in the ../Custom directory of ANSYSEM_DIAG_RESULTS_DIR directory.
6. Run the diagnostics script from a submit host for the cluster

Environment variables

The following environment variables are applicable for both Linux and Windows environment.

ANSYSEM_DIAG_PROD_DIR

Environment variable	ANSYSEM_DIAG_PROD_DIR
Description	Location of the Ansys EM installation. This must be available at the same path from all the hosts of the cluster.
Windows example	`\\filer\AnsysEM\v231\Win64`
Linux example	`/shared/ansysem/v231/Linux64`
Comments	Windows: Required. Linux: Optional. Export this environment variable if you make a copy of the diagnostic script.

ANSYSEM_DIAG_RESULTS_DIR

Environment variable	ANSYSEM_DIAG_RESULTS_DIR
Description	Location of the diagnostic report and other results on a shared drive. This must be available at the same path from all the hosts of the cluster
Example	`\\filer\Home\User\Ansoft\HPCDiag`
Linux example	`/shared/home/user/Ansoft/HPCDiag`
Comments	Windows: Required. Linux: Optional. Export this environment variable if the home directory for the user is not accessible from the cluster.

ANSYSEM_DIAG_CUSTOM_DIR

Environment variable	ANSYSEM_DIAG_RESULTS_DIR
Description	Location of the configuration of product tests and other custom site-specific tests. This location must be on a shared drive that is available at the same path from all the hosts of the cluster
Example	`\\filer\Home\User\Ansoft\HPCDiag\Custom`
Linux example	`/shared/home/user/Ansoft/HPCDiag/Custom`
Comments	Windows: Optional. You may want to specify it if the path %ANSYSEM_DIAG_RESULTS_DIR%\..\Custom is not suitable Linux: Optional. Export this environment variable if the home directory for the user is not accessible from the cluster.

How the diagnostic tool works

The diagnostics are run as a scheduler managed job. Running the diagnostic script submits a scheduler job that runs the diagnostic tool on the hosts allocated to the job. Once the diagnostic job starts, the tool executes a set of diagnostic tests. These tests run on each host allocated to the job, collecting diagnostic information relevant for running HPC jobs. The tool combines the diagnostic information to produce an HTML report. The tool saves the HTML diagnostic report and other results in a shared drive, which must be available at the same path from all the hosts of the cluster. On Linux, the default is Ansoft/HPCDiag subdirectory under the user's home directory. On Windows, the user must specify this location using the ANSYSEM_DIAG_RESULTS_DIR environment variable.

Related Topics

High Performance Computing (HPC) Integration

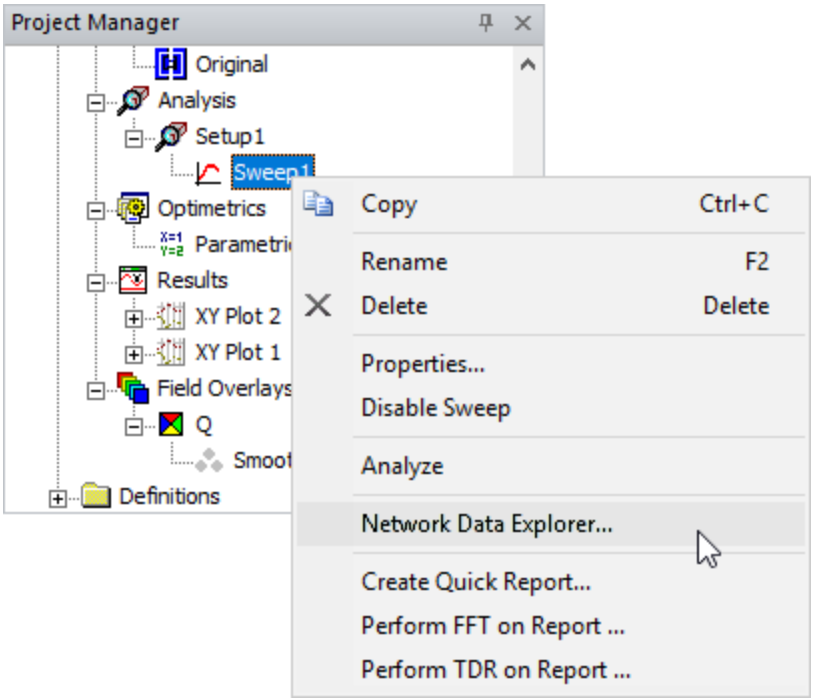
22 - Network Data Explorer

The Network Data Explorer provides visualization, analysis, and manipulation tools for network data of 3D and 2D Eddy Current solutions.

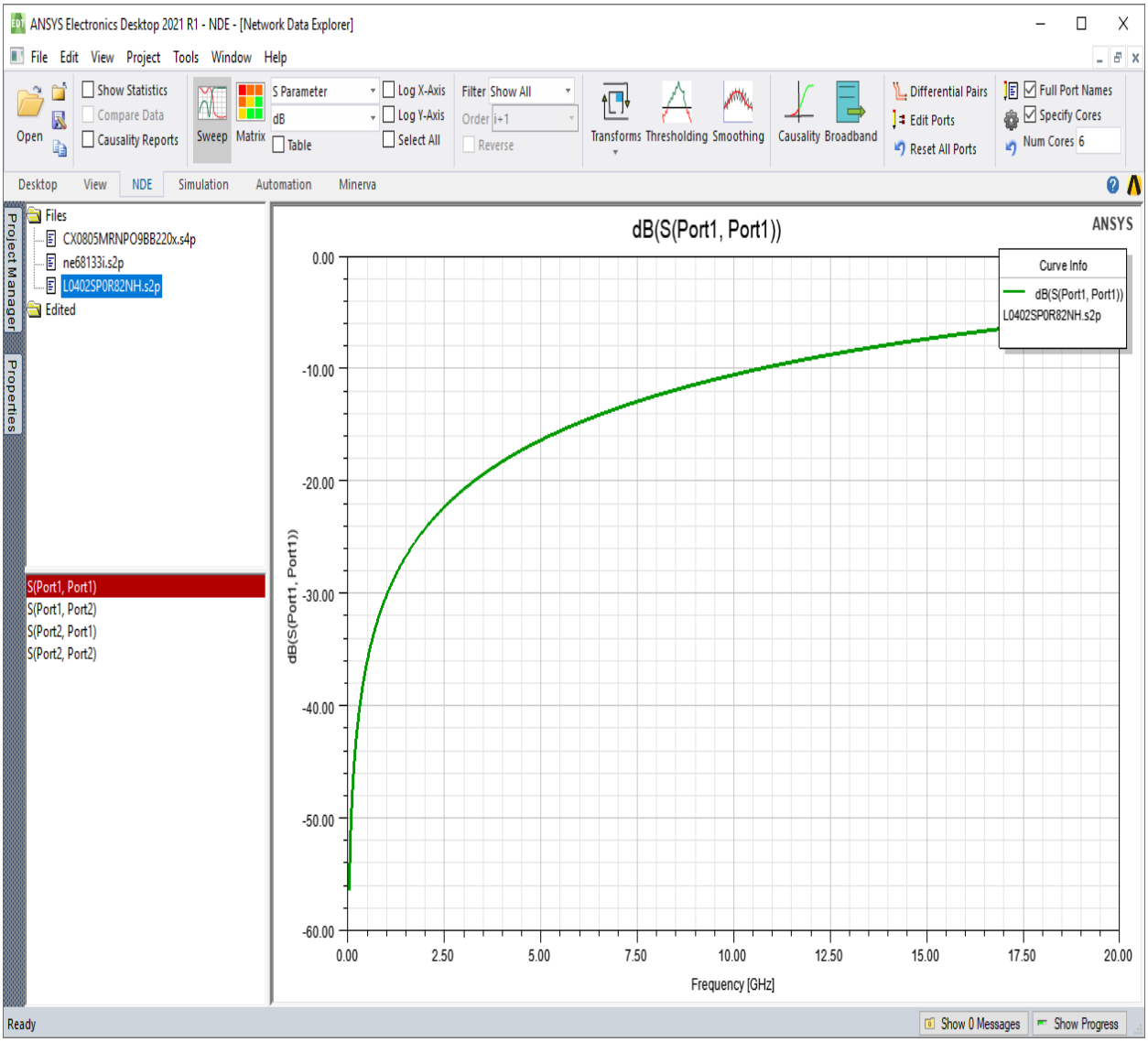
To access Network Data Explorer, click **Tools>Network Data Explorer**.

If you have run a simulation with data that can be converted into network data, you can also access Network Data Explorer through the shortcut menu in the Project Manager.

Note: Accessing Network Data Explorer from this menu automatically loads simulation data.



The **Network Data Explorer** window appears.



Related Topics

Network Data Explorer Overview

Loading Data into Network Data Explorer

Exporting Data from Network Data Explorer

Network Data Explorer Ribbon

Data Display Pane Context Menus

Exploring Network Data and Modifying the Display

Causality Checking and Plots

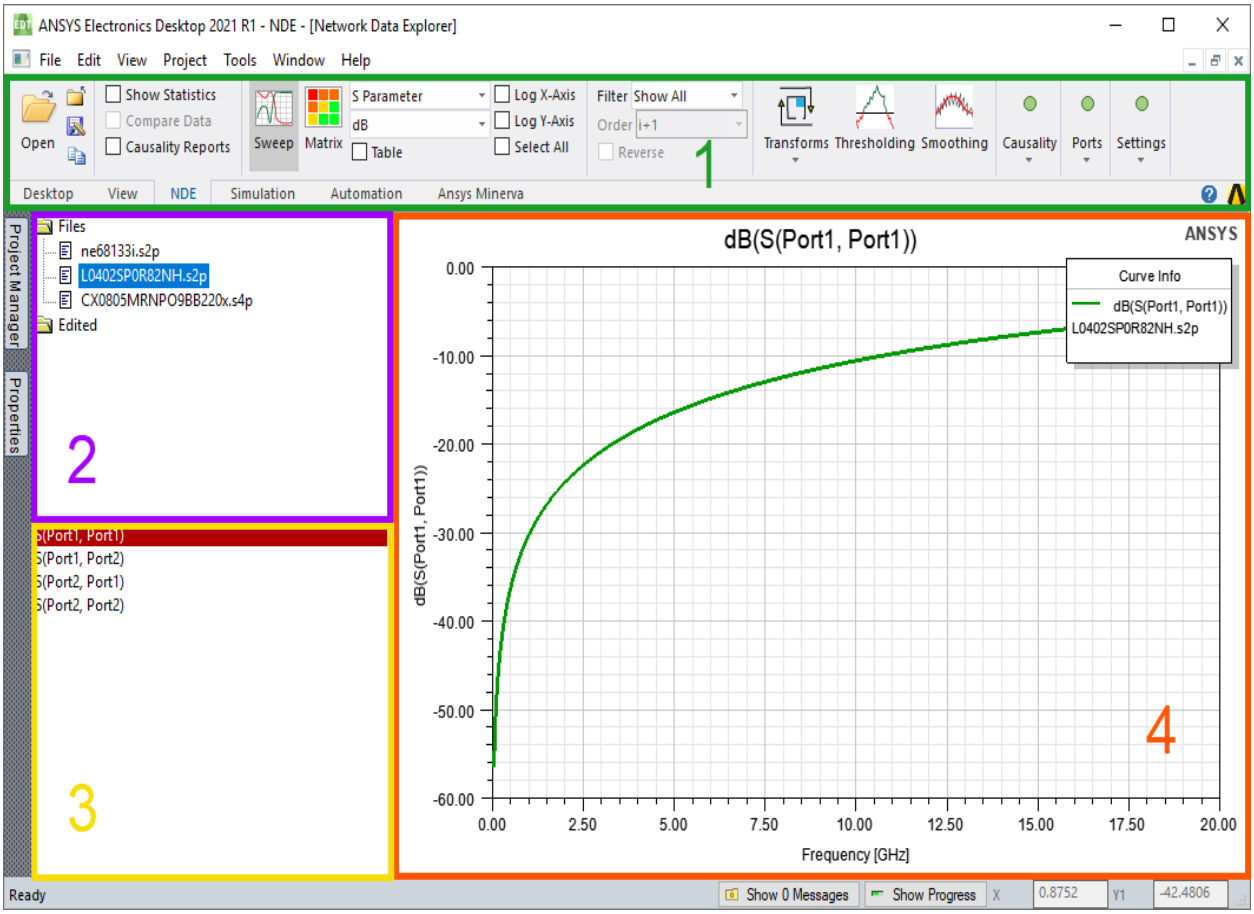
Multithreading

Network Data Explorer Overview

The Network Data Explorer window is divided into the following panes:

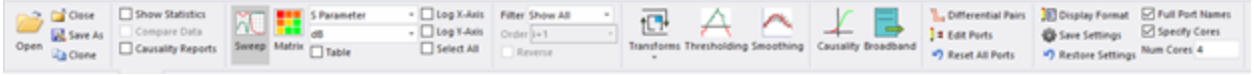
1. **NDE ribbon** – allows you to perform many functions of the Network Data Explorer.
2. **Network Data Selection pane** – allows you to select a network data file.
3. **Cell and Frequency Selection pane** – allows you to narrow your selection.
4. **Data View pane** – displays data in table or plot format.

The panes are shown in the following figure. Additional information about each pane follows.



NDE Ribbon

The ribbon provides access to many of the Network Data Explorer's functions and display options.

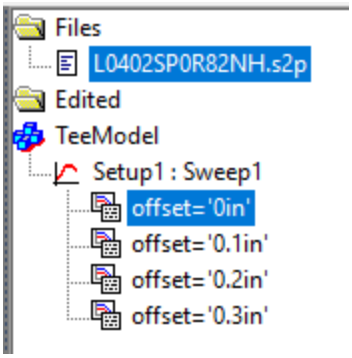


On this ribbon, you can control:

- Plotting - allows you to determine how the data is displayed.
- Quantity – allows you to select the type of quantity to display (parameter values, matrix statistics, or causality plots).
- Parameter Type – allows you to choose the parameter for display (S, Y, or Z parameters, Port Impedance, or Gamma).
- Format – allows you to decide the display function to apply to the data (e.g., magnitude, phase, dB, real, imaginary).
- Export – allows you to export either SYZ data (*.s1p, *.ts, *.nmf, *.tab, *.m, *.cit) or Broadband data (*.sp).
- Check – allows you to check causality.
- Cores – allows you to enable or disable multithreading.
- Post-Process Selection – allows you to choose between Terminal Data and Differential Pairs, if your design includes Differential Pairs

Network Data Selection Pane

This pane allows you to view and compare various data sets. Original data sets appear under Files. Click one of these to see the data set as it was when it was opened. Altered data sets are listed under Edited. These data sets appear here when they have been smoothed, transformed, or changed in some way.



This pane also lists available variations for AEDT design solution data. Variations are listed under a setup name icon (for example “Setup1 : Sweep1”) that is listed under a design icon (“TeeModel”). Variations can be selected and displayed just like other data sets.

Use the **Shift** and **Ctrl** keys to select and display data sets from multiple files simultaneously. Multiple data sets can be selected to display multiple traces in sweep plots, but not in tables. Click the **Files** icon to select all data sets under **Files**. Click the **Edited** icon or a design/setup icon to do the same.

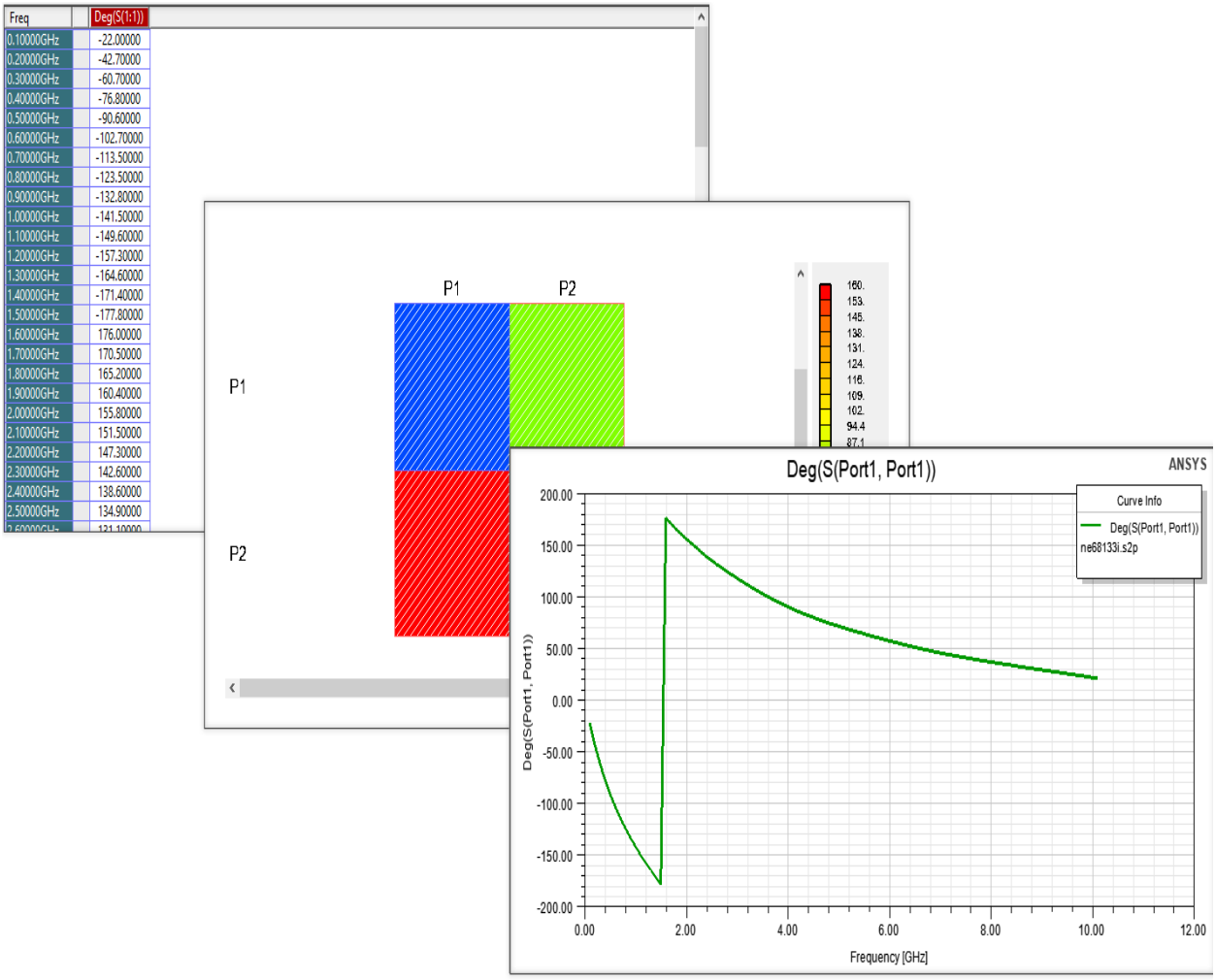
Cell and Frequency Selection Pane

Select the **Sweep** button on the NDE ribbon to display cell entries, for example S(1,1). Select the **Matrix** button to display frequencies. When displayed by frequency, the entire matrix is presented in the Data View pane for each selected frequency. When displayed by matrix cell, the data for the individually chosen cells is shown across all frequencies. Use the **Select All** check box to select all frequencies or cells.

> **Note:** For Maxwell 2D solutions when the geometry is in XY mode, a Model Depth selection is available. The default Matrix selection is the alphabetically first defined matrix parameter. The default Model Scale setting is 1. The default Model Depth setting is 1 meter.

Data View Pane

The Data View Pane displays data for **Sweep** or **Matrix** in either plots or a table, depending on your selection.



Loading Data Into Network Data Explorer

If you launched **Network Data Explorer** using the shortcut menu, the current solution data is automatically loaded and ready for viewing. Otherwise, you must load a data file into Network Data Explorer.

You can import the following file types:

- Touchstone Format (*.s*p)
- Touchstone 2 Format (*.ts)
- Citifile (*.cit)
- Neutral Format (*.nmf)
- State Space File (*.sss)

Note	When this type of file is loaded, Network Data Explorer regenerates s-parameter data based on the file. You can compare the regenerated s-parameters to the original data.

To import a file into Network Data Explorer, either drag and drop an analysis from the Project Manager into Network Data Explorer or:

1. On the **NDE** ribbon, click **Open**. An **Open** window opens.
2. Navigate to and select a file.
3. Click **Open**. The file appears in the **Files** tree.

The file browser allows you to open multiple files at a time. However, the displayed data always corresponds to the data set indicated in the Network Data Selection pane. Click the file you want in that pane to switch between data sets.

Exporting Data from Network Data Explorer

Network Data Explorer allows you to export data to a variety of different file formats.

In this section, you will learn about:

Exporting SYZ Data

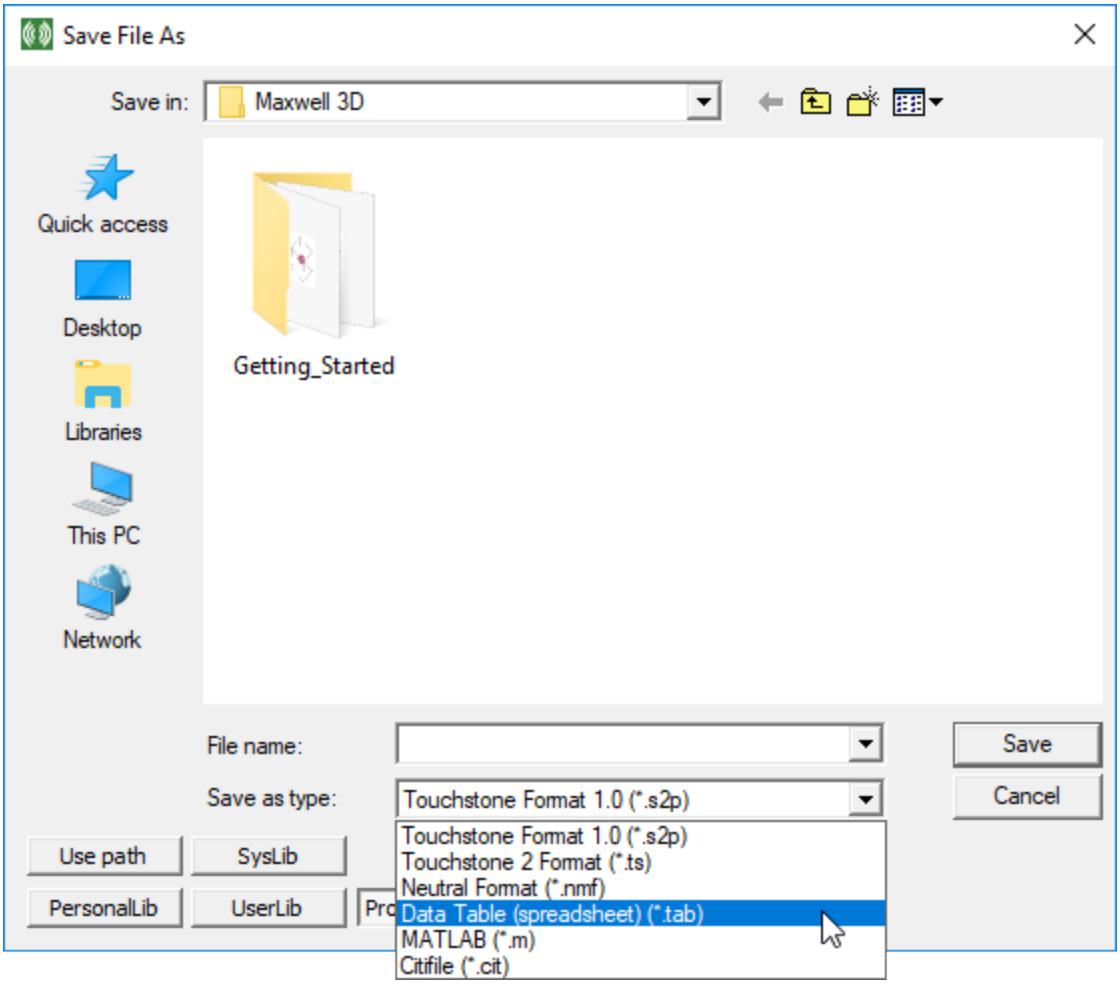
Exporting Macro Model

Creating an NPort Model

Scripting for Network Data Explorer

Export SYZ Data

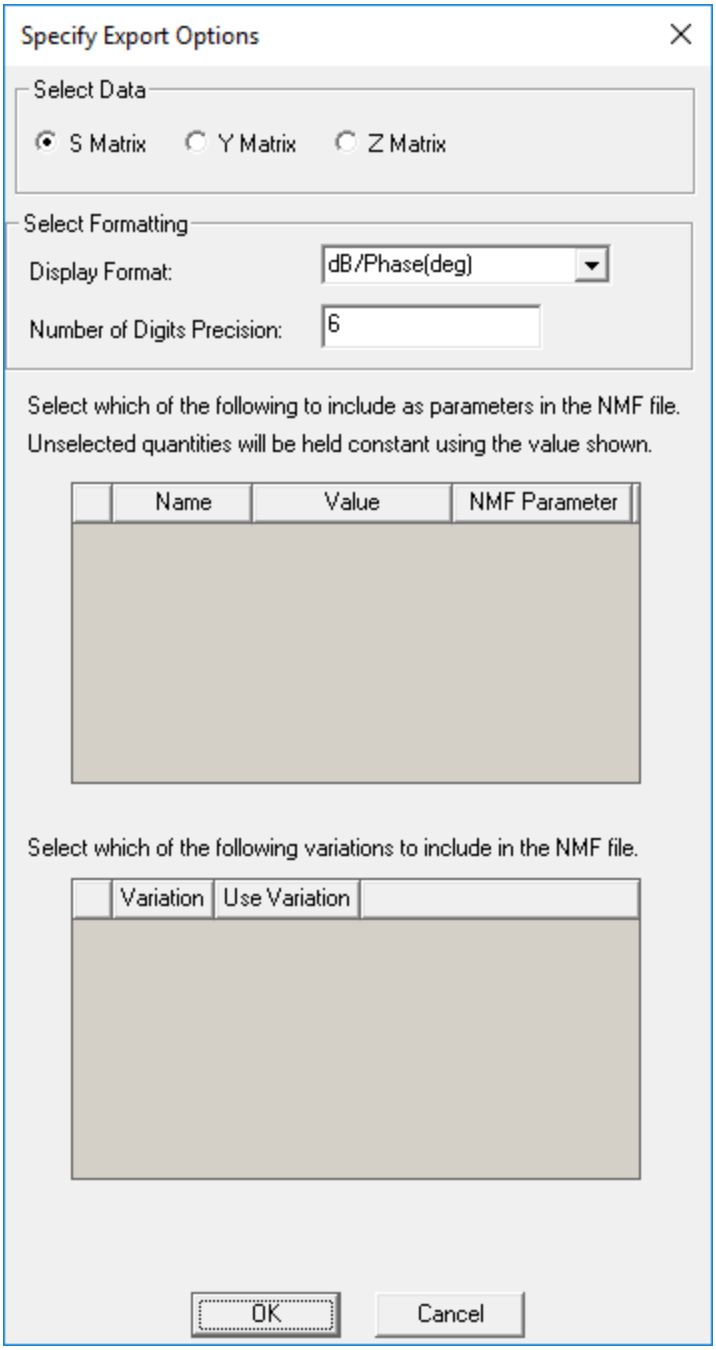
To export SYZ data from within Network Data Explorer, click the **Save As** icon on the **NDE** ribbon. The **Save File As** window appears.

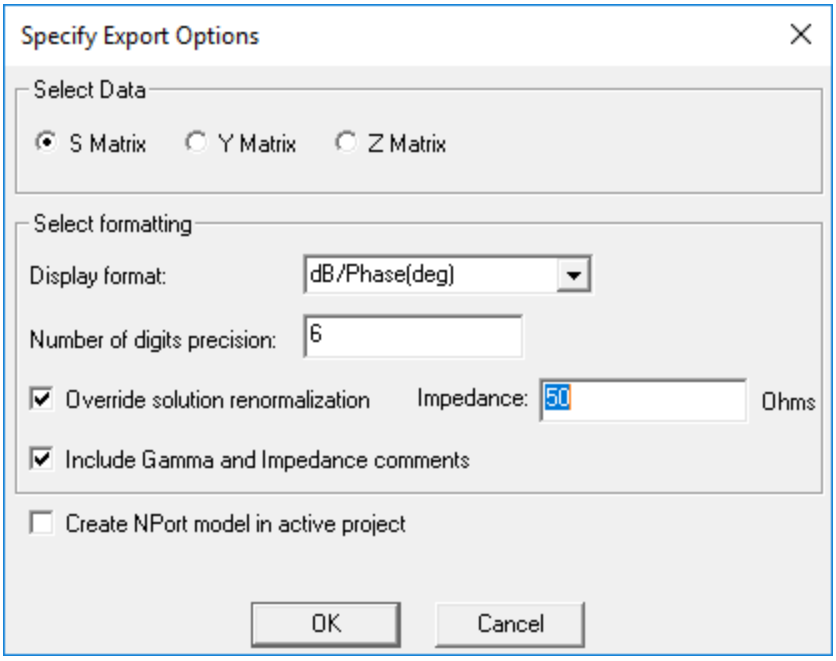


You can export data in any of six file types:

- Touchstone Format 1.0 (*.s*p)
- Touchstone 2 Format (*.ts)
- Neutral Format (*.nmf)
- Data Table Spreadsheet (*.tab)
- MATLAB (*.m)
- Citifile (*.cit)

Select a file type and name for export. A **Specify Export Options** window appears.





Depending on the type of export file, different options appear. However, all file types allow you to:

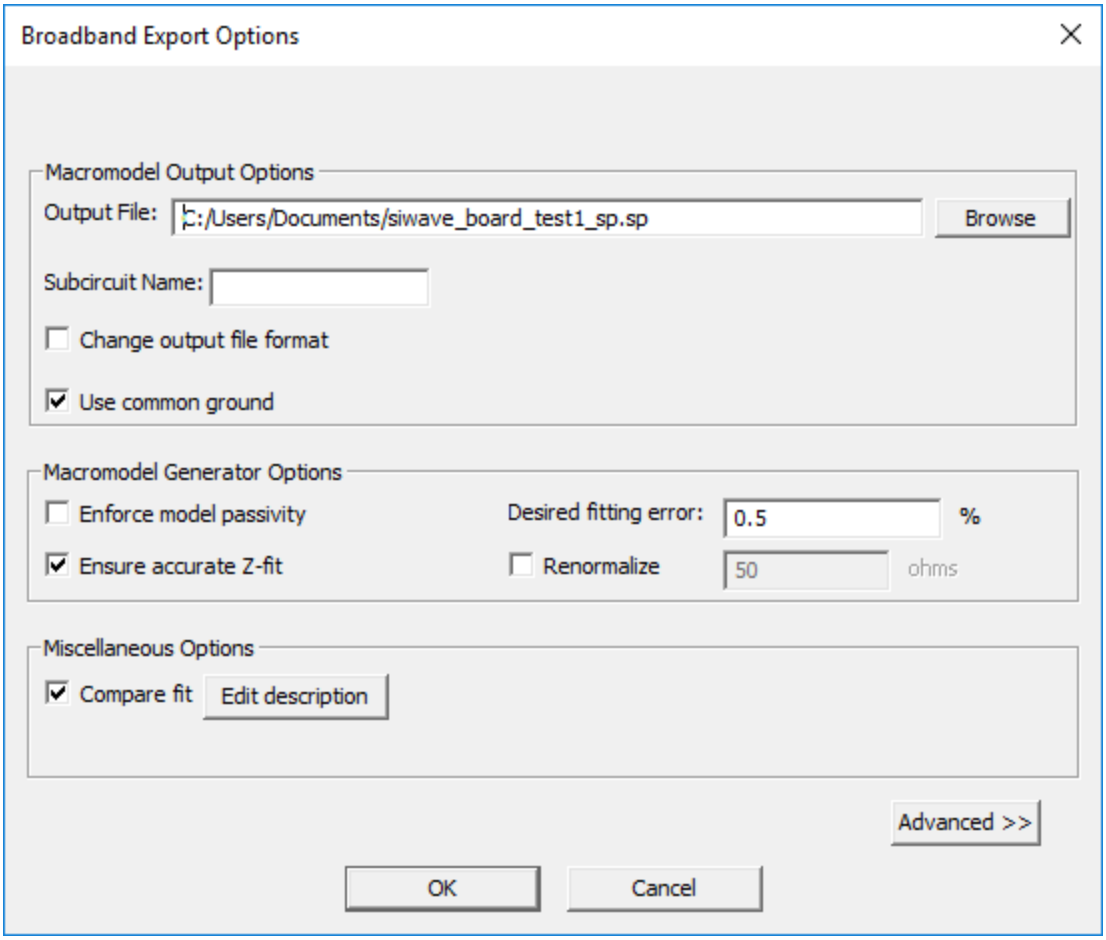
- Select from **S Matrix**, **Y Matrix**, and **Z Matrix** data.
- Select the **Display Format**.

Some types allow you to create an NPort Model in the active project.

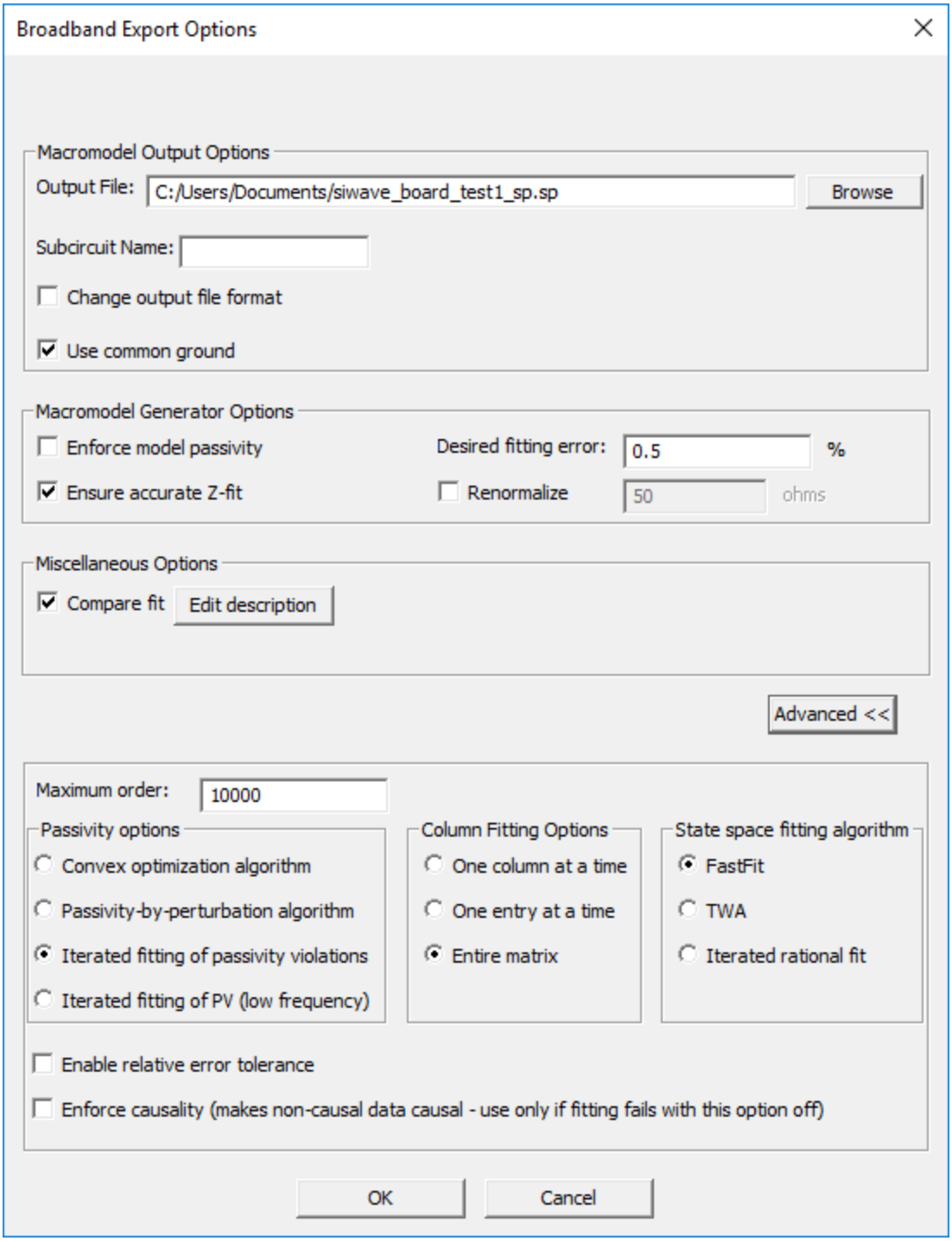
Exporting Macro Model

To export data, click the **Broadband** icon on the **NDE** ribbon.

The **Broadband Export Options** window appears.



Click **Advanced >>** to view all options.



Macromodel Output Options include:

- **Output File** – Allows you to choose the name and location of the file.
- **Subcircuit Name** – Use this field to name the subcircuit.

- **Change Output File Format** – Check this box to open a submenu allowing you to select a new output format.
- **Use Common Ground** – Check this box to use common ground. When this option is on, ports are referenced to ideal ground (node 0). When this option is off, extra ports are generated to provide the reference levels. Common grounding is best when the pins are physically near to each other and ideal ground is suitable. For distant connections and circuits with non-ideal reference levels such as differential pairs, common grounding is not used.

Note	◦ R and L values may be quite sensitive to the values of the S-parameters. This is an issue if the actual impedance value is much greater than or much less than the reference impedance of the S-parameters. ◦ Since resistances of power cables is typically in the milliohms range at DC, using a reference impedance of 50 ohms is 5000 times higher. This causes any fitting errors in the state space model to get multiplied by 5000 times when the R and L values are computed. ◦ As a general rule, for high power applications a reference impedance of 1 ohm is probably a better choice than 50 ohms.

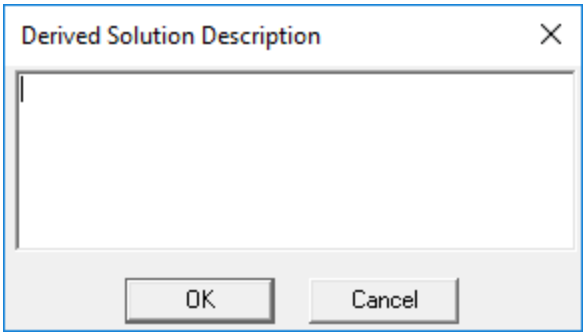
Macromodel Generator Options include:

- **Enforce Model Passivity** – Check this box to enforce passivity.
- **Ensure Accurate Z-fit** – Check this box when state-space fitting of Y-parameters or Z-parameters does not produce an accurate fit.
- **Desired Fitting Error** – Allows you to select the value at which rational fitting fails (if the fitting error exceeds this value).
- **Renormalize** – Check this box to renormalize using the specified impedance item. 50 ohms is the default setting, but you can type a different value.

Note	• R and L values may be sensitive to S-parameter values. This presents an issue if the actual impedance value is much greater than or much less than the reference impedance of the S-parameters. • Since resistances of power cables are typically in the milliohms range at DC, using a reference impedance of 50 ohms is 5000 times higher. This causes any fitting errors in the state space model to be multiplied by 5000 when the R and L values are computed. • For high-power applications, a reference impedance of 1 ohm is generally a better choice than 50 ohms.

Miscellaneous Options include:

- **Compare Fit** – When this box is checked, the original and derived solution will be available for comparison. You can click **Edit Description** to open the **Derived Solution Description** window and add a text description to better identify the export.



Advanced Options include:

- **Maximum Order** – Allows you to specify the number of poles. See Note below.
- **Passivity Options** – If you enabled **Enforce Model Passivity**, this area allows you to select the passivity enforcement method. **Iterated fitting of passivity violations** (IFPV) is the default method. The other options are methods that can be tried if IFPV yields a poor fit. The **Iterated fitting of PV (low frequency)** option builds upon the existing IFPV with the objective of improving the fit to "Z" at DC and low frequencies.
- **Column Fitting Options** – This area allows you to choose how poles are matched to columns:
  - **One Column at a Time** – The set of poles will be shared across all entries of a single column.
  - **One Entry at a Time** – Each entry will be fitted using a separate set of poles.
  - **Entire Matrix** – The set of poles will be shared across all entries of the matrix being fitted.
  -

Notes	• Typically, using the same set for all entries is adequate, and yields the most compact models. However, if all the entries of the matrix have completely unrelated transfer functions, it may be better to fit them using separate pole sets. • The options **One column at a time** and **One entry at a time** do not work when either **Ensure accurate Z-fit** or FastFit is used.

- **State Space Fitting Algorithm** – Allows you to select FastFit, TWA, Iterated rational fitting. See Note below.
  - **FastFit** (default) – FastFit is the Ansys-proprietary method for state-space fitting. Network Data Explorer uses FastFit for calculating the state-space matrices from the network data. The FastFit algorithm for state-space fitting is an alternative to the Tsuk-White algorithm (TWA) and Iterated Rational Fitting (IRF) methods. FastFit is generally as accurate as TWA, but is significantly faster than both TWA and IRF. It also aims to fit the lower frequencies with higher fidelity.

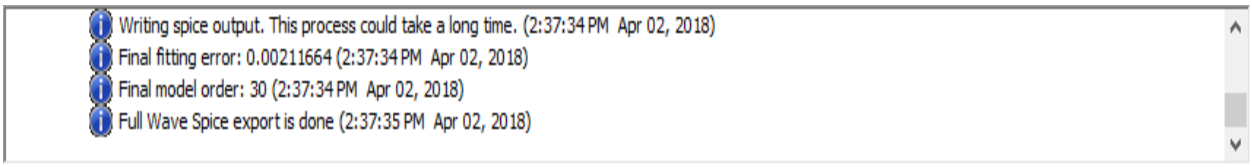
  - **TWA** – The Tsuk-White Algorithm is an Ansys-proprietary method for fitting a state space model to extracted s-parameter data. It uses techniques based on Singular Value Decomposition (SVD) to quickly determine required number of poles for fitting a model.
  - **Iterated Rational Function** – The IRF fitting approach takes a matrix of S-parameter data and, for each matrix entry, tries a succession of different pole-zero approximations (increasing the number of poles used at each iteration) until it can find an acceptable fit to the data. For broad frequency sweeps and large numbers of excitations, this process can be time consuming because of all the iterations and is not guaranteed to produce a good fit to the data. It is retained as a fallback if the TWA algorithm fails.
- **Enable Relative Error Tolerance** – Allows you to enable relative error tolerance, which works best with TWA fitting.

Notes	Note that the **Enable Relative Error Tolerance** option works best with the TWA fitting algorithm, is not recommended for use with iterated rational fitting, and is disabled when either FastFit or **Ensure accurate Z-fit** is used.

- **Enforce Causality** – Allows you to make non-causal data causal. Use this option only if fitting fails without it.

Notes	Broadband models are built from a rational-function approximation to the data. The fidelity of this approximation can be controlled by setting the Maximum order (number of poles).

Click **OK** to begin the export. The **Messages** pane details the export process.



Comparing Original S-Parameters with Exported S-Parameters

If **Compare Fit** was checked during export, the **Data Selection** pane updates to list both the original and exported solution *and* the **Compare** checkbox is checked.

Creating an NPort Model

Touchstone, Neutral Format, and Citifile exports allow you to select an option to **Create NPort Model in active project**.

Creating an NPort Model exports the active data set back to the design, either as a static n-port model or as a parametric n-port model. If there are multiple variations, a parametric n-port model

is automatically created containing all variations. When only a single variation exists, the user can create a static n-port model that either links to a file (the data is exported first) or stores the data itself. Linking to a file reduces the size of the ADSN file. The newly created model can be placed as a component in a circuit. This is particularly useful after reducing the number of ports via termination.

The import of the exported solution is done by reading the exported Touchstone file.

Scripting for Network Data Explorer

Scripting is available for each Network Data Explorer export method, which means a script can be recorded to duplicate the export process.

Network Data Explorer can be invoked in the following contexts, and scripting is available from the Project Context and the Design Instance Context (simulation setup):

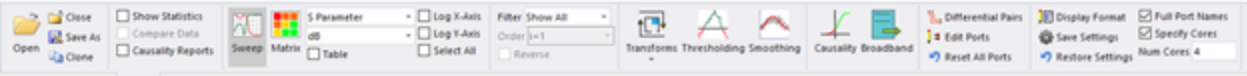
- **Project Context** – In the context of a project, you can open a touchstone file and then export.
- **Design Instance Context** – In the context of a solution (RCM from the simulation setup in a design), you can export the corresponding network data solution.

When there is no design available, export functionality and scripting are not available.

For more information, see Network Data Explorer Script Commands in the *Ansys Electronics Desktop Scripting Guide*.

Network Data Explorer Edit Menu Commands

The **Edit** menu on the **Control** pane includes several commands.



In this section, you will learn about:

Data Sources

Setting Display Format

Displaying Full Port Names

Saving or Resetting Default Settings

Magnitude Thresholding

Smoothing All Frequencies

Cell Filtering

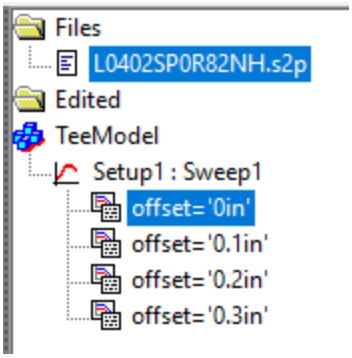
Editing Port Properties

Defining Differential Pairs and Displaying Mixed Mode Parameters

Resetting All Port Properties

Network Data Explorer Data Sources

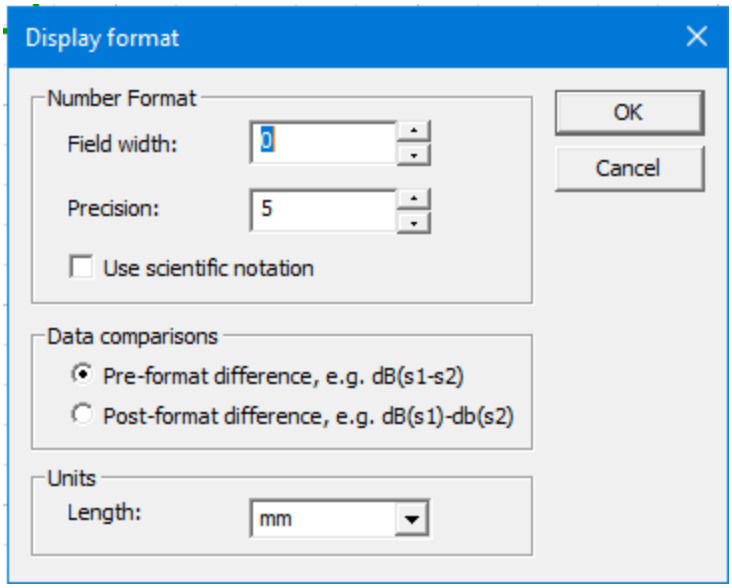
Network Data Explorer allows you to easily view all data sources in the **Network Data Selection** pane.



Network Data Explorer Display Format

The **Display Format** window affords additional control over the display of values in Network Data Explorer.On the **NDE** ribbon, click the **Display Format** icon.

The **Display Format** window appears.

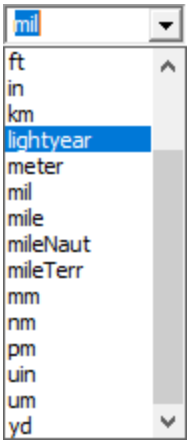


The **Number Format** options allow you to specify **Field Width** (the minimum number of characters used to display a number) and **Precision** (the number of decimals to display). You can also check the **Use scientific notation** checkbox, if desired.

The Data Comparisons options allow you to choose Post-format difference or Pre-format difference.

- **Post-format difference** – when comparing data sets, subtract values after applying the formatting function (e.g. dB, magnitude); the values displayed will be the difference between the magnitude, dB, and so on.
- **Pre-format difference**- when comparing data sets, subtract values before applying the formatting function (e.g. dB, magnitude); the values displayed will be the magnitude, dB, etc., of the complex difference.

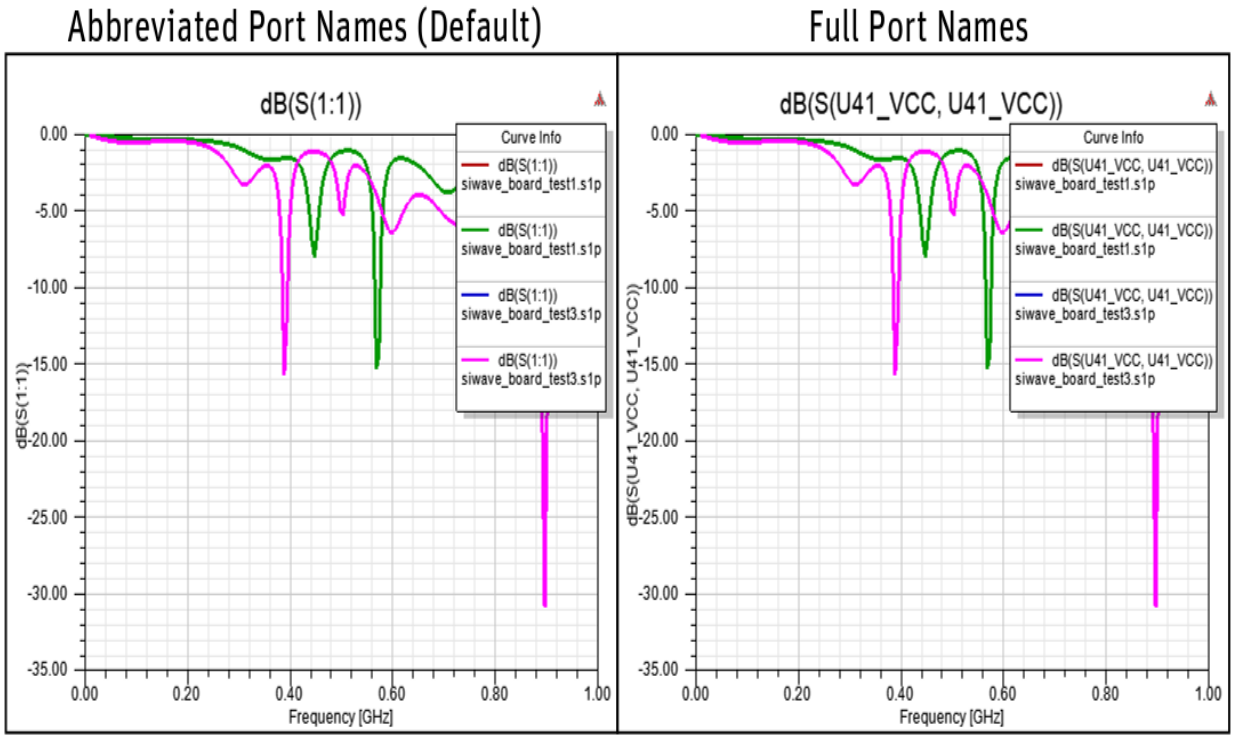
The **Units** option allows you to specify the **Length** unit (the unit used to display and interpret length values). The default is mm.



Network Data Explorer Display Full Port Names

By default, full port names are displayed. This applies to both the Data Selection pane and the Data View pane. To change this so that port names in Network Data Explorer are displayed in an abbreviated form (P1, P2, etc.), click the **Full Port Names** check box on the **NDE** ribbon.

The following figure shows the difference in display for a plot.



Note	Tool-tips always display the full port name.

Network Data Explorer Save or Reset Default Settings

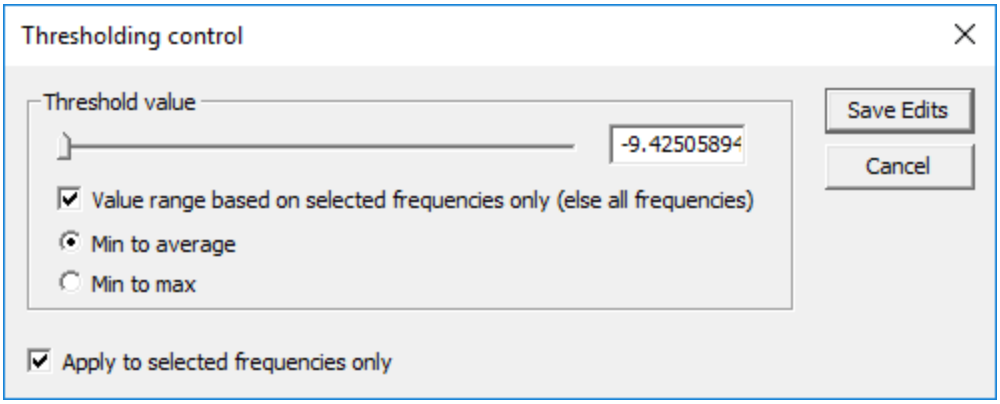
To save field settings as the default, click the **Save Settings** icon on the **NDE** ribbon. The next time Network Data Explorer is opened, the chosen settings will be selected by default.

To restore the default settings (i.e. the settings previously saved using the **Save Settings** button), click the **Restore Settings** icon on the **NDE** ribbon. The next time Network Data Explorer is opened, the original settings will be selected by default.

Network Data Explorer Magnitude Thresholding Control

Thresholding is a way of reducing matrix entries to zero based on a value threshold. Thresholding applies to the displayed data (e.g., Magnitude), but the original matrix value is reset to zero. To access threshold settings, click the **Thresholding** icon on the **NDE** ribbon.

The **Thresholding Control** window appears.



Set the threshold value by sliding the control or by entering a value. The data range for the slider is determined by the options below:

- **Value range based on selected frequencies only** – When selected, the slider value range is determined from the selected frequencies only, otherwise, the entire data set is used to determine the range.
    - **Min to average** – When selected, the slider data ranges extends from the data minimum to the mean.
    - **Min to max** – When selected, the slider data range extends over the entire data range, from minimum to maximum data values.
- **Apply to selected frequencies only** – When this box is checked, the **Save Edits** button only modifies the currently selected network data matrices. Otherwise, the matrix data for all frequencies and variations is adjusted.

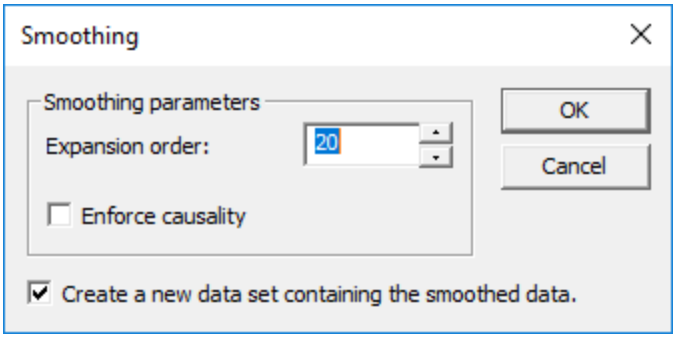
Click **Save Edits** to apply thresholding values.

Smoothing

To access data smoothing options:

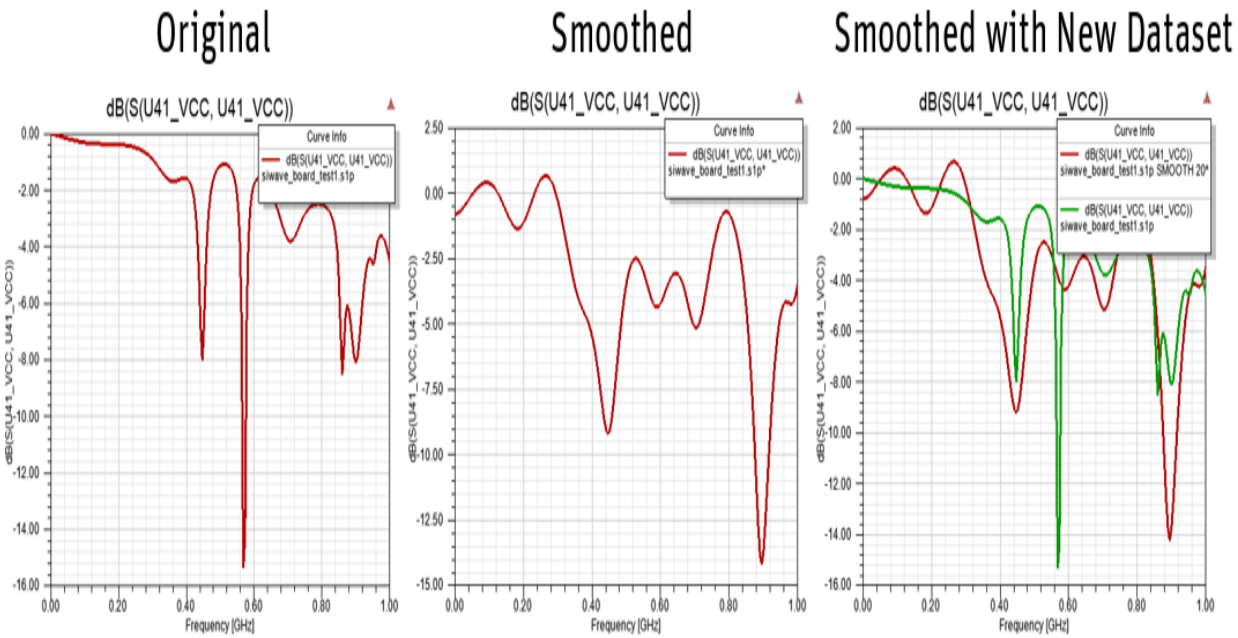
1. Click the **Smoothing** icon on the **NDE** ribbon.

The **Smoothing** window appears.



2. The **Smoothing Parameters** area allows you to choose the **Expansion Order**. This can be any discrete value between 1 and 150.
3. If desired, check the **Enforce Causality** checkbox.
4. If desired, check the **Create a new data set containing the smoothed data** checkbox. If selected, the smoothed data appears alongside the original data.
5. Click **OK**.

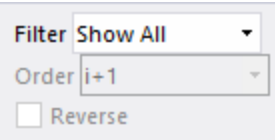
The Data View Pane updates. The following image shows a plotted result with **Create a new data set containing the smoothed data** unchecked and checked.



A least-squares polynomial fit of the specified order is used to interpolate new data points for the magnitude and phase components of the S-parameters.

Cell Filtering

The cells available in the data selection pane may be restricted using cell filtering. The **Cell Filtering** control are located on the **NDE** ribbon.



Cell filtering is modeless, and filters are immediately applied to the cell list. Filtering remains in effect when the window has been closed.

For an n-port model with a total of 2n pins in the standard arrangement, the choices are:

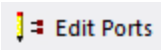
- **Show all** – display all available cells. There are n-squared choices.
- **Return loss** – show S(i, i). There are n choices.
- **Insertion loss** – show S(i, i+1). There are n choices.
- **Lower triangle** – show S(i, j) for all j < i. There are n(n-1)/2 choices.

Three pin arrangements are recognized:

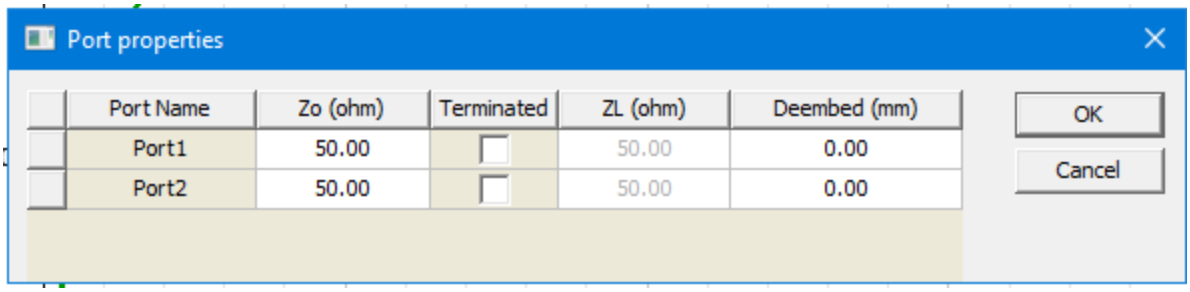
- **S(i, i+1)** and its **Reverse order**, S(i, i-1)
- **S(i, i+n)** and its **Reverse order**, S(i, i-n)
- **S(i, 2*n-i+1)** and its **Reverse order**, S(i, 2*n-i-1)

Changing Port Properties and Reducing Matrix Size

The normalization impedance, termination, port order, gamma values, and de-embedding distance may all be edited from the **Port Properties** window.



To access these options, click **Edit Ports** on the **NDE** tab. The **Ports Properties** window opens.



Ports appear in a table. Click a column heading to sort by that column. Click within a cell to edit the port property:

- **Zo (ohm)** and **ZL (ohm)** – specify Impedance values. Accepted syntaxes are:
  - real (e.g., 50)
  - real + imag i (e.g., 50+5i)
  - imag i (e.g., 5i).
- **Terminated** – use the checkbox to terminate a port. Terminated ports are eliminated from the matrix, reducing the matrix size. Existing data sets with mismatching port numbers will no longer be available for data comparisons.
- **De-Embedding** – this column appears only if gamma values are available. Default units can be changed from the Set Display Format window.

To reorder ports, click and drag a row to a new location.

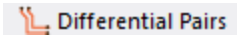
To save changes, click **OK**.

Displaying Mixed-Mode Parameters using Differential Pairs

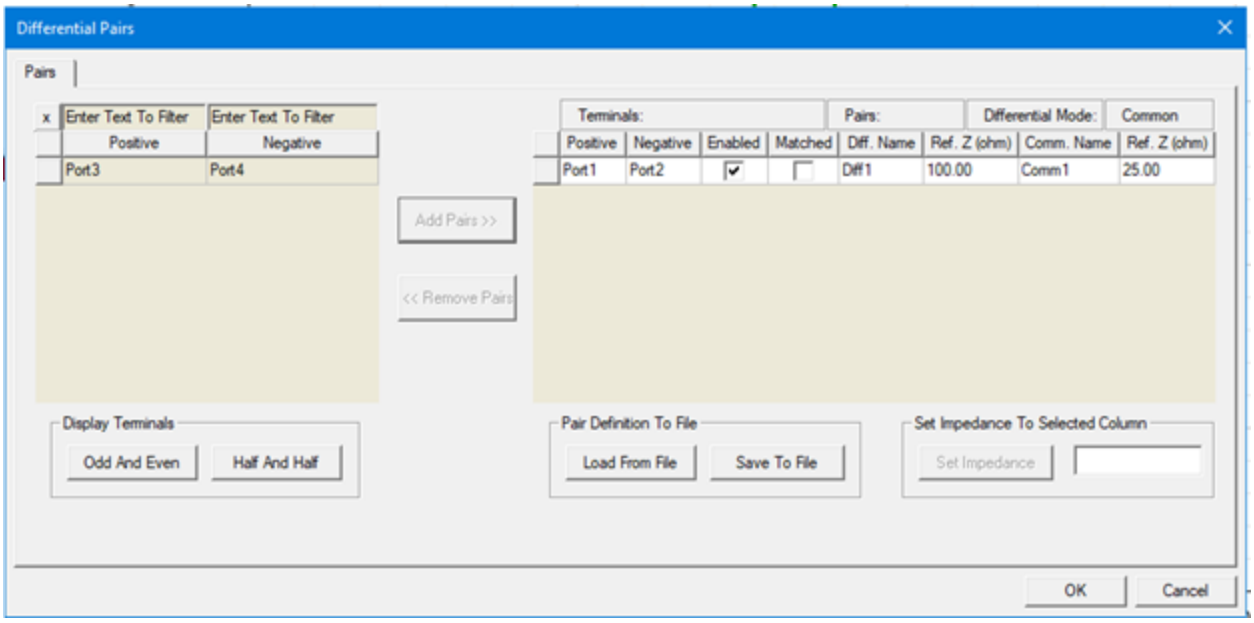
Network Data Explorer displays mixed-mode parameters when differential pairs are both defined and activated.

To define differential pairs:

1. Select existing ports.
2. Open Network Data Explorer (**Tools>Network Data Explorer**).
3. On the **NDE** ribbon, select **Differential Pairs**.



The **Differential Pairs** window appears.



4. Select a pair from the list on the left and click **Add Pairs**.
5. Click **OK**.
6. To disable all differential pairs, click the **Enabled** column header in the **Differential Pairs** dialog to deselect all pairs.

Note	The Network Data Explorer **Edit** menu option **Reset All Port Properties** deactivates all pairs, but it does not clear the differential pair settings. And since **Reset All Port Properties** also clears reference impedances and terminations, it should not be used when the user simply wishes to disable all differential pairs.

Reset All Port Properties

Reset All Ports resets all changes in the **Edit Ports** window. It also deactivates all differential pairs defined in the **Differential Pairs** window but does not remove the definitions.

Data View Pane Context Menus

The Data Display pane presents different right-click menu options, depending on the context. Some of the commands are the same as those on the NDE ribbon. Others only appear in the context menus.

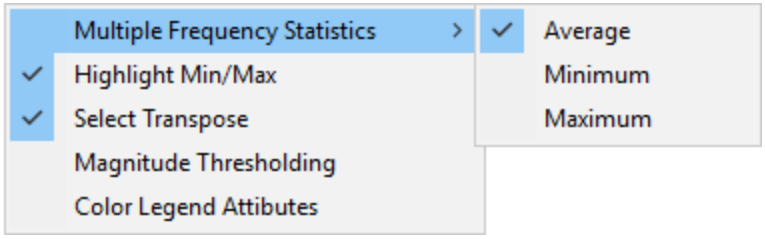
Network Data Explorer commands unique to the context menus are:

- Multiple Frequency Statistics
- Highlight Min/Max
- Select Transpose
- Color Legend Attributes
- Matrix Entries Plot Menu

Multiple Frequency Statistics

The **Multiple Frequency Statistics** menu option determines the statistical composite to display when multiple frequencies have been selected for the matrix display. The statistical data is always the first matrix displayed, followed by matrices for each individual frequency. The **Multiple Frequency Statistics** option also indicates the data used in the colored matrix plot when multiple frequencies have been selected.

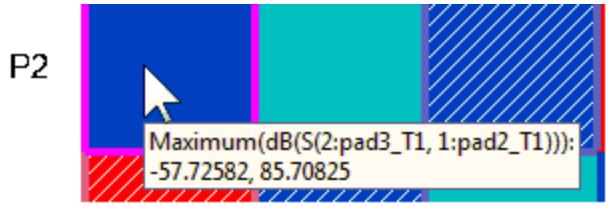
This right click menu option appears in the **Matrix** plot, regardless of whether you are in Table or Plot view.



The menu options are:

- **Average** – display the average of the matrix values across selected frequencies.
- **Minimum** – display the minimum matrix values across selected frequencies.
- **Maximum** – display the maximum matrix values across selected frequencies.

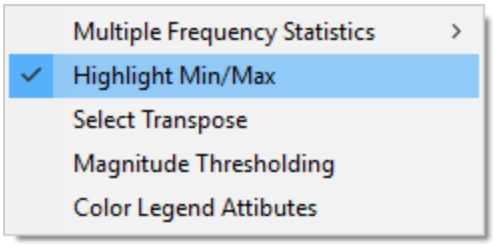
The selected information also appears in a tool-tip when you hover the cursor over a cell.



Highlight Min/Max

The **Highlight Min/Max** menu option determines whether the minimum and maximum matrix entries should be highlighted in the matrix table and color plot view.

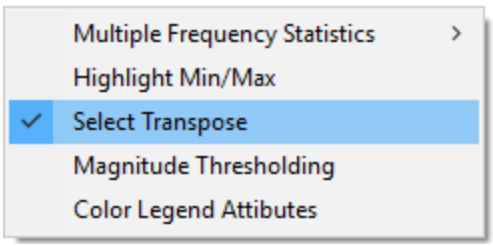
This right click menu option appears in the **Matrix** plot, regardless of whether you are in Table or Plot view.



Select Transpose

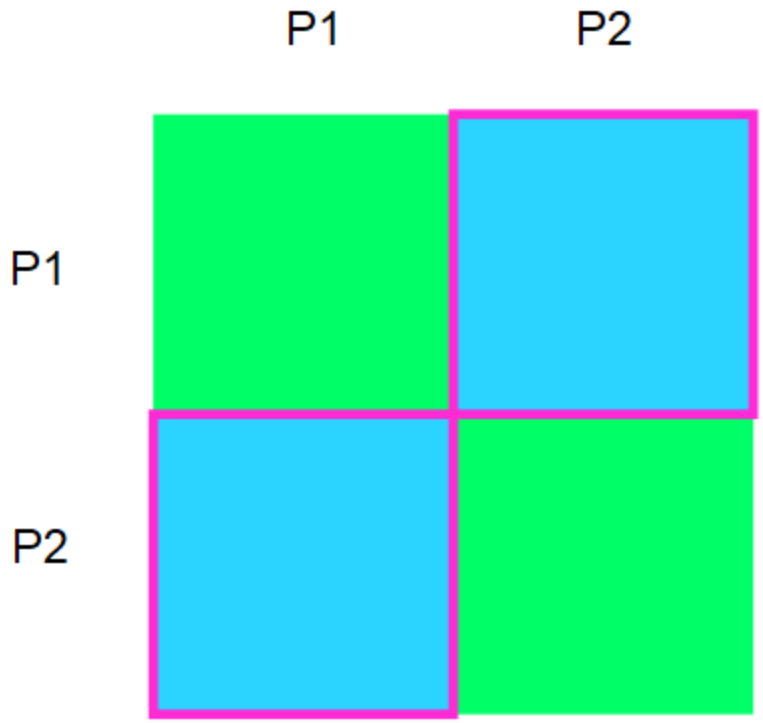
The **Select Transpose** menu option determines whether transpose cells are highlighted along with selected cells.

This right click menu option appears in the **Matrix** plot, regardless of whether you are in Table or Plot view.



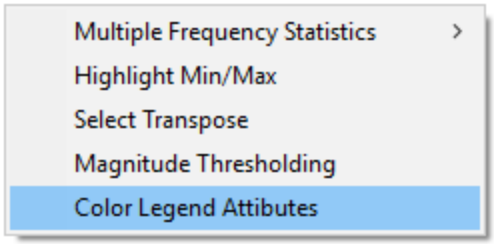
Transpose cells are highlighted in Table or Plot view, as shown below.

0.01000GHz	P1	-57.10340, -81.97841	-57.72581, 85.70836	-0.00213, -0.18916
	P2	-57.72582, 85.70825	-57.09556, -81.98749	-66.27113, 99.16916
	P3	-0.00213, -0.18916	-66.27110, 99.16941	-57.10246, -81.95892
	P4	-66.27112, 99.16942	-0.00213, -0.18929	-57.72556, 85.71465

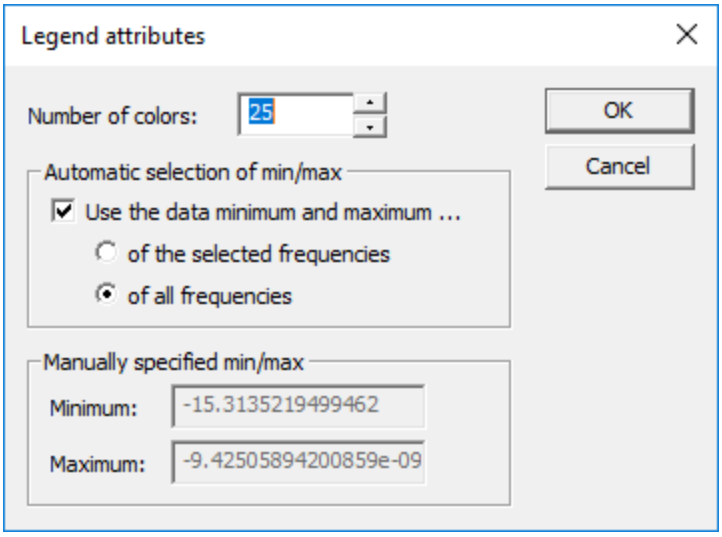


Color Legend Attributes

The **Color Legend Attributes** menu option allows you to change the granularity of the color scheme and the value range for plots. This right click menu option appears in the **Matrix** plot.



Alternatively, double-click the matrix color plot to open the **Legend Attributes** window.



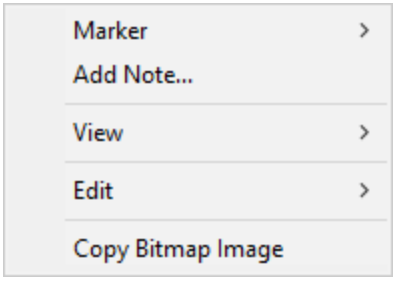
Options include:

- **Number of colors** – allows you to select number of color entries in the legend (the number of divisions between the start/end of the data range). This can be set to any discrete number between 1 and 50.
- **Automatic selection of min/max** – check the **Use the data minimum and maximum** checkbox to automatically select the data range using the minimum and maximum values from either selected frequencies or all frequencies in the data set.
- **Manually specified min/max** – when the when the range is not automatically determined, these fields permit the user to manually enter hard values. For example, for S parameter data magnitude data, you could enter a minimum of 0 and a maximum of 1.

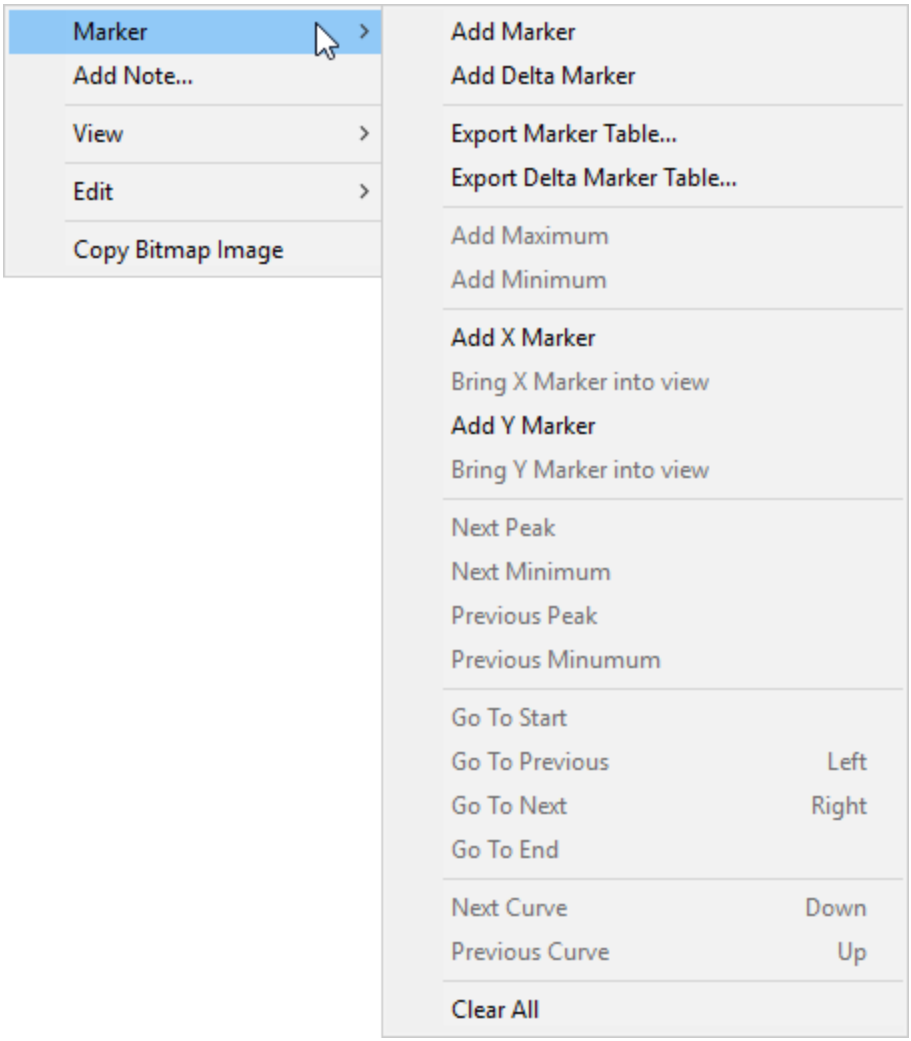
Using a standard range across all frequencies permits you to quantitatively compare plots, and ndExplorer remembers legend settings for each data-type and display-format pair.

Matrix Entries Plot Menu

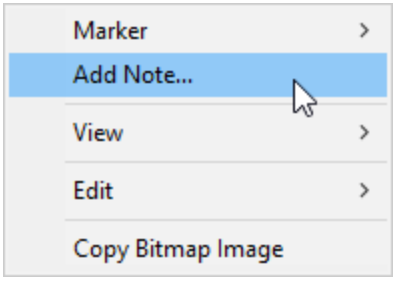
When the **Matrix entries** radio button is selected and you are viewing the data in **Plot** mode, several new right-click menu options appear.



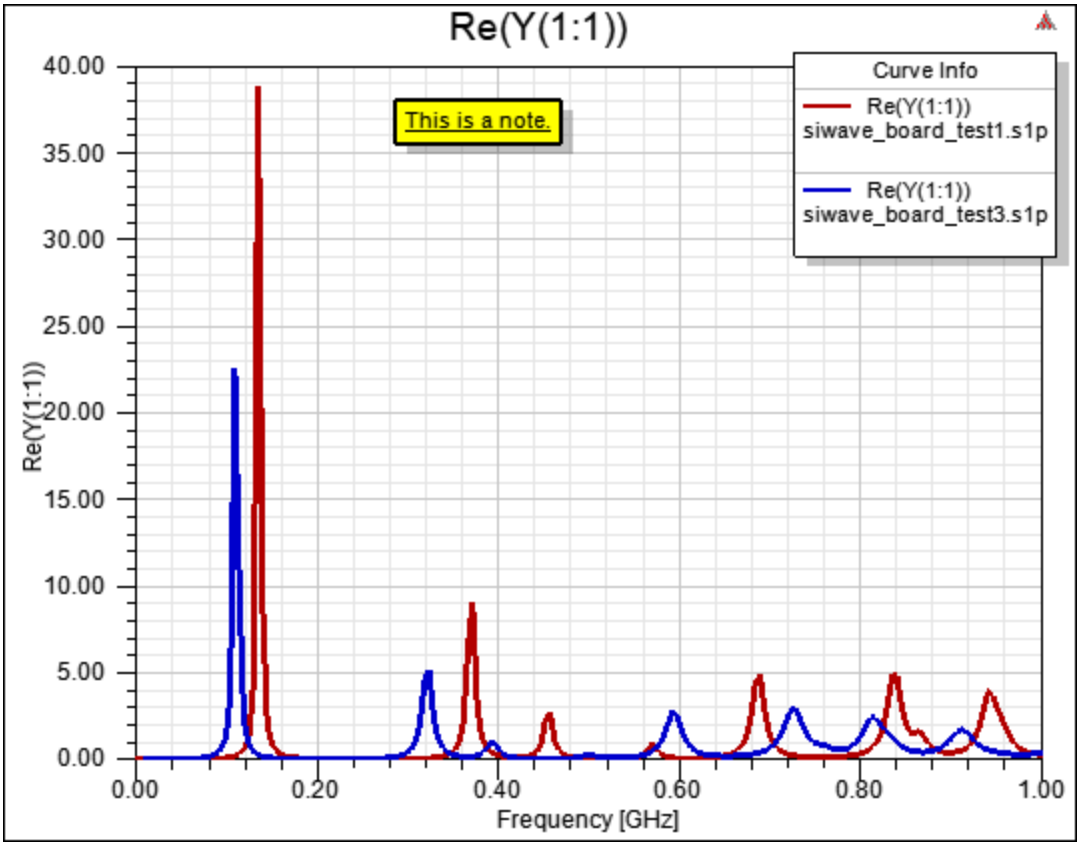
The **Marker** sub-menu provides commands for adding markers to plots.



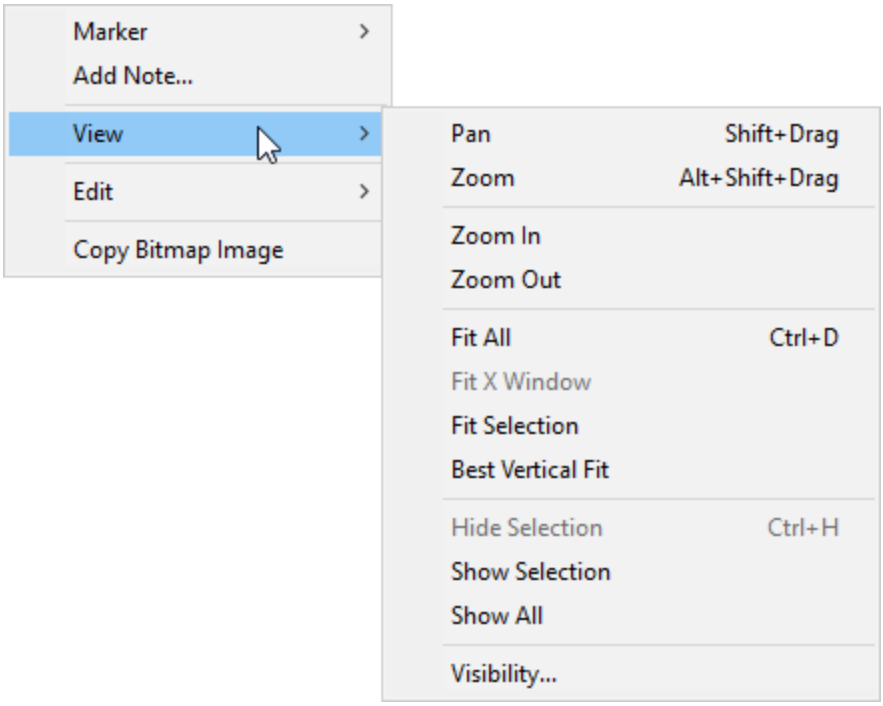
The **Add Note** menu option allows you to add a note to the plot.



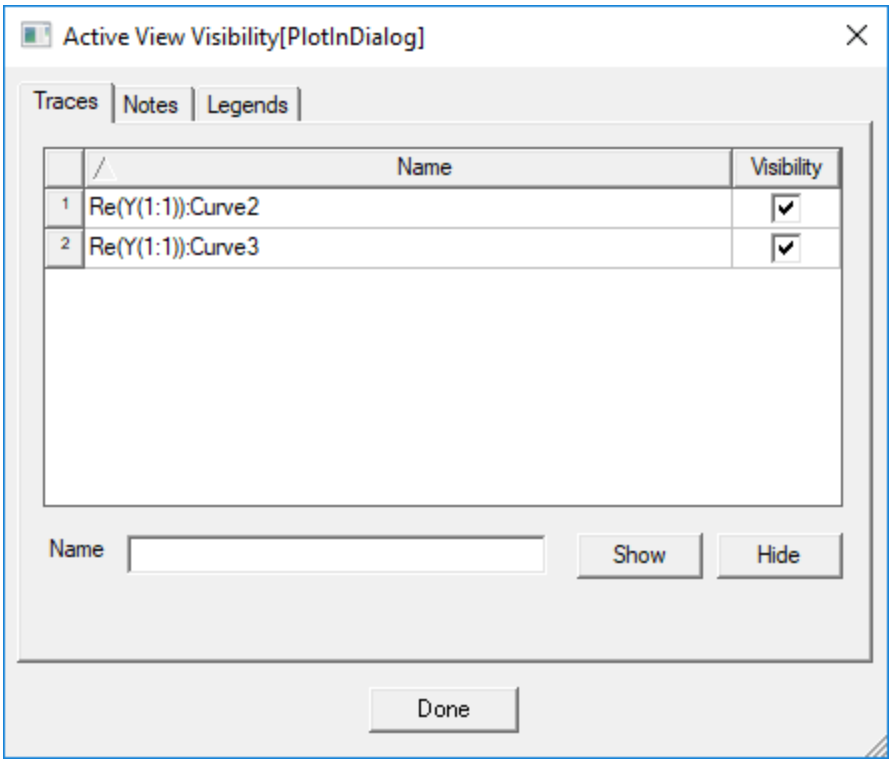
The note appears at the location you right-clicked. You can click and drag the note to a new location, or double-click the note to change its color and font.



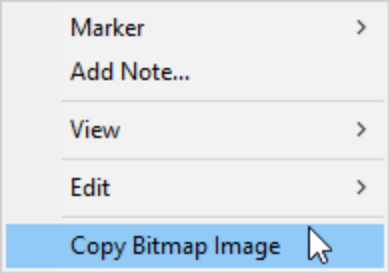
The **View** sub-menu provides commands for viewing, panning, zooming, and fitting elements to your plot.



Click **Visibility** to open the **Active View Visibility** window, where you can select the visibility of traces, notes and legends.



The **Copy Bitmap Image** menu option copies the plot to your clipboard. You can then paste it into a graphics editor.



Exploring Network Data and Modifying the Display

Network Data Explorer allows you to view data and modify various aspects of the display, including color plots, color coding, viewing across frequencies, and displaying individual statistics. This section provides examples of Network Data Explorer capabilities:

This section provides examples of Network Data Explorer capabilities:

- Viewing the S, Y, or Z Matrix for a Frequency
- Viewing a Color-coded Matrix Plot
- Displaying a Cell Graph Across All Frequencies
- Displaying Matrix Statistics by Frequency
- Displaying Individual Statistics for All Frequencies
- Creating a Statistics Plot
- Comparing Network Data

Related Topics

Magnitude Thresholding

Smoothing

Cell Filtering

Editing Port Properties

Displaying Differential Pairs as Post Process Selection

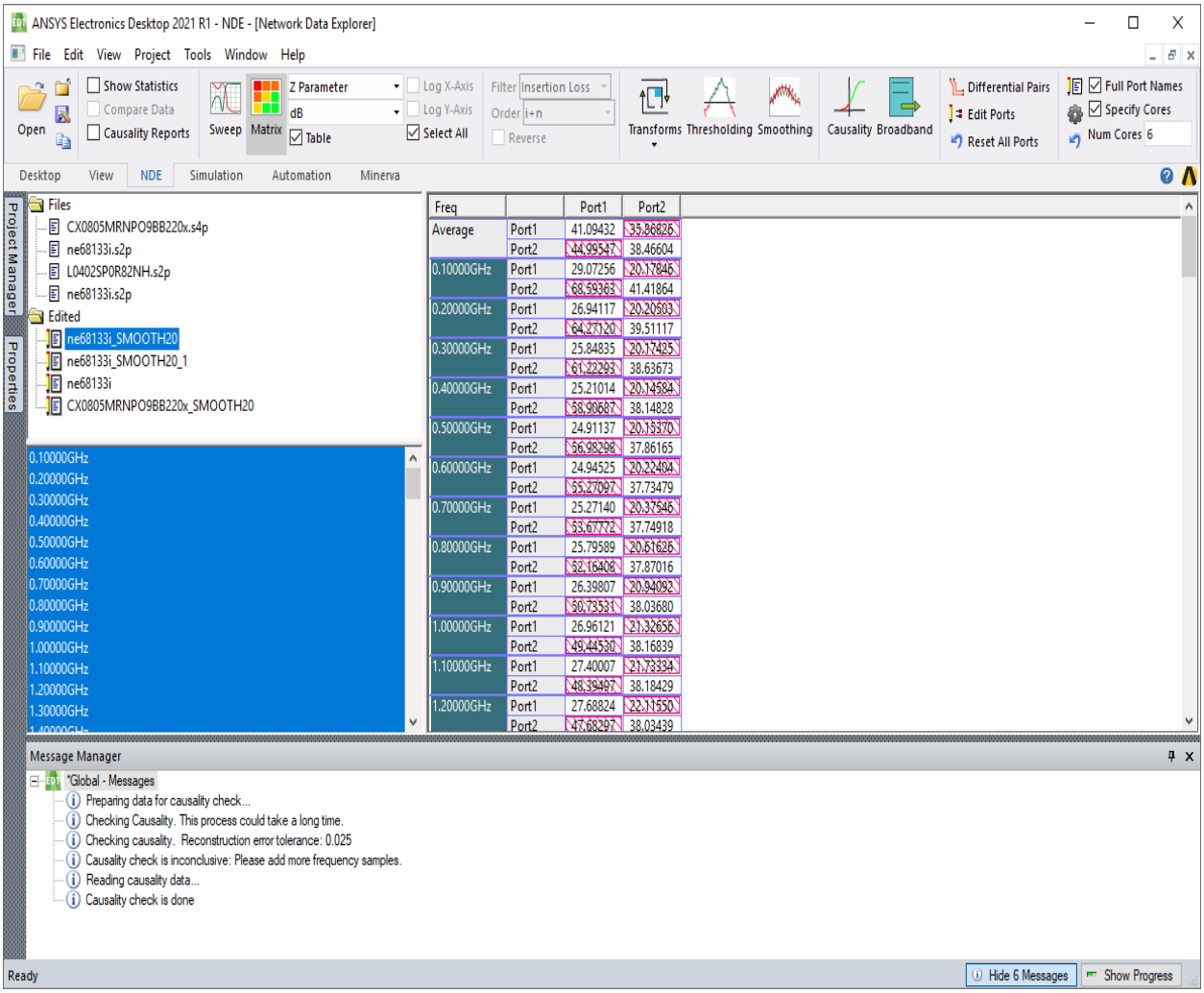
Viewing the S, Y, or Z Matrix for a Selected Frequency

To view the S, Y, or Z matrix:

1. On the **NDE** ribbon, click **Matrix**
2. Click the **Table** check box.
3. Use the **Parameter** type drop-down menu to select **S parameter**, **Y parameter**, **Z parameter**, or another choice.
4. Choose a format from the drop-down menu, for example: dB or Mag.

5. In the Cell and Frequency Selection pane, select frequencies to display or click the **Select All** check box on the **NDE** ribbon to select all frequencies.

   The S, Y, or Z matrix displays.



6. Hover over a cell to see more information in a tool tip.
7. Right click to see a right click menu with other commands.

Maximum values are highlighted in red stripes. If Select Transpose is enabled, transposes are highlighted in red stripes as well.

Minimum values are highlighted in blue stripes.

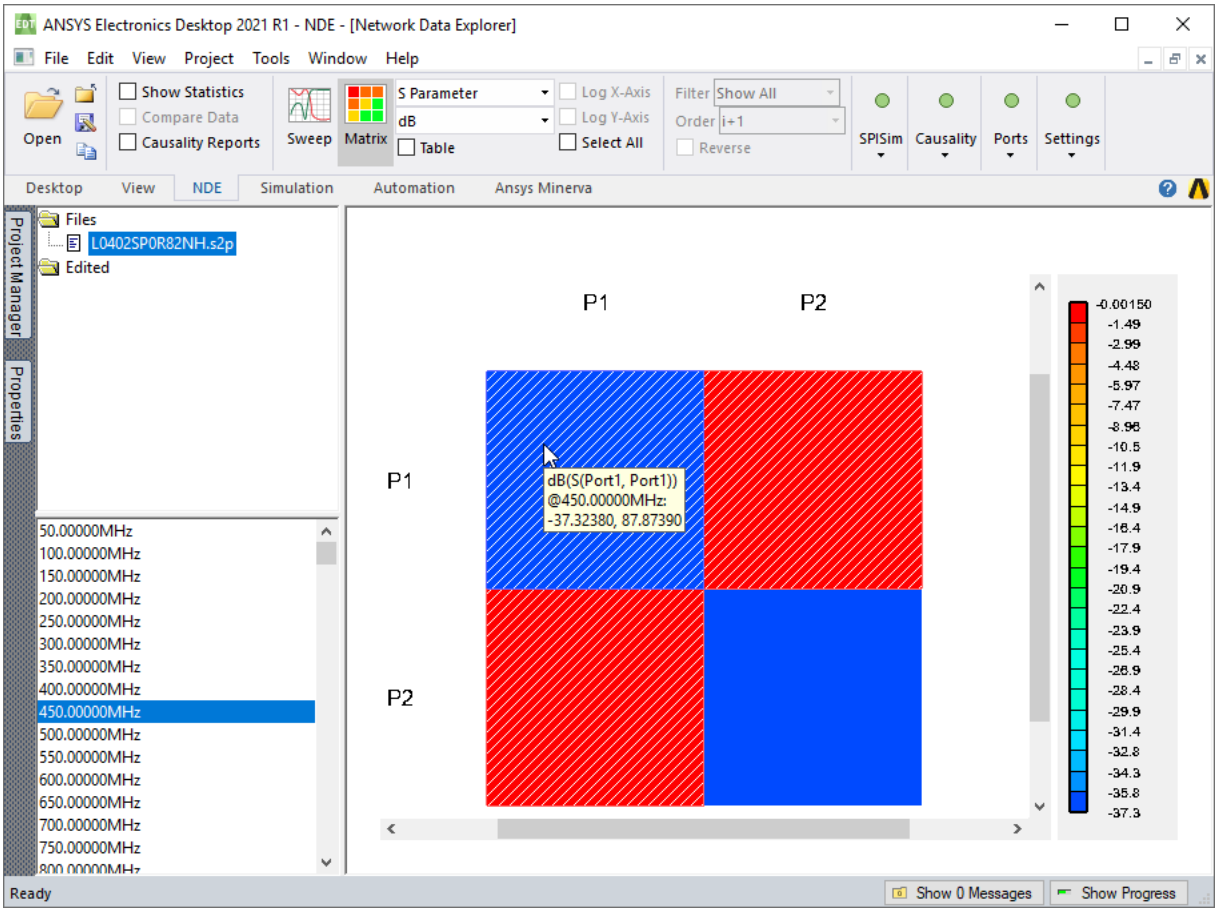
Selected cells appear in solid blue. If Select Transpose is enabled, the transpose is highlighted in blue as well.

Double-clicking a cell switches to a matrix cell view, in which values for all frequencies for that cell are displayed. The double-clicked frequency is highlighted with solid red shading

Complex values are compared using their modulus. When multiple frequencies or variations are selected, the data display depends on the Multiple Frequency Statistics setting.

Viewing a Color-coded Matrix Plot

1. On the **NDE** ribbon, click **Matrix**.
2. Use the **Parameter** type drop-down menu to select **S parameter**, **Y parameter**, **Z parameter**, or another choice.
3. In the Cell and Frequency Selection pane, select the frequencies you want to plot or click the **Select All** check box on the **NDE** ribbon to select all frequencies.
4. Hover over a cell to see more information in a tool tip.
5. Right click to see a right click menu with other commands.



Matrix values display in a color-coded grid. If the selected **Format** is a complex value, only the real component is used to determine the display color. When multiple frequencies or variations are selected, the data display depends on the Multiple Frequency Statistics setting. Maximum values are highlighted in red. Minimum values are highlighted in dark blue.

Hover the cursor over any cell to view information about it.

Click any cell to select it. Selected cells appear with a pink outline. If Select Transpose is enabled, the transpose is selected in pink as well.

Double-click any cell to view a matrix cell plot in which all frequency values for that matrix cell are displayed as a graph.

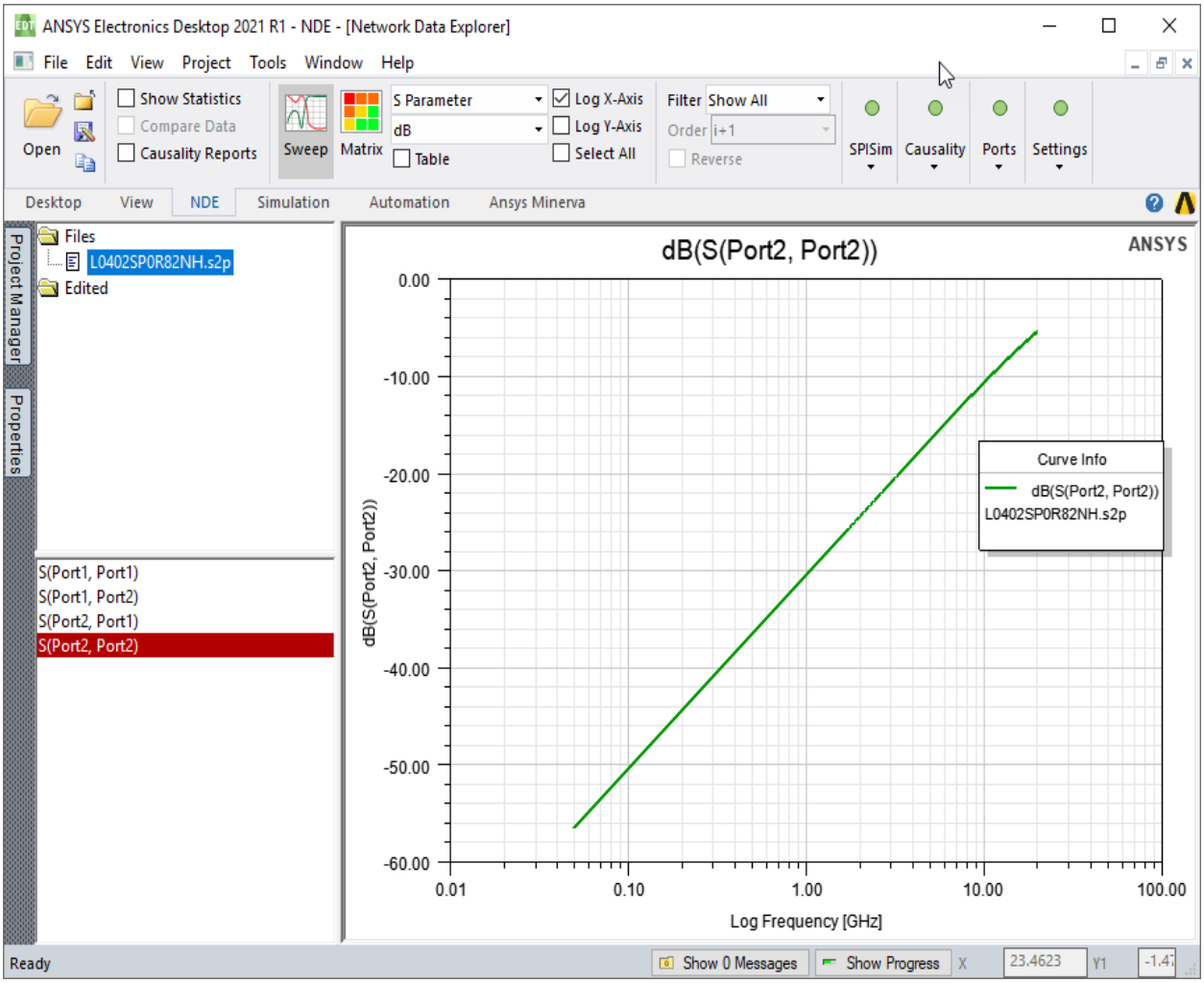
Related Topics:

- Setting Multiple Frequency Statistics
- Changing Color Legend Attributes

Displaying a Cell Graph Across All Frequencies

Network Data Explorer can plot a cell across all frequencies.

1. On the **NDE** ribbon, click **Sweep**. The Data View Pane updates if necessary.
2. Use the **Parameter** type drop-down menu to select **S parameter**, **Y parameter**, or **Z parameter**.
3. In the Cell and Frequency Selection pane, select cells to display or click the **Select All** check box on the **NDE** ribbon to select all cells.
4. To add a log scale to the X-axis, click **Log X-Axis** on the **NDE** ribbon.



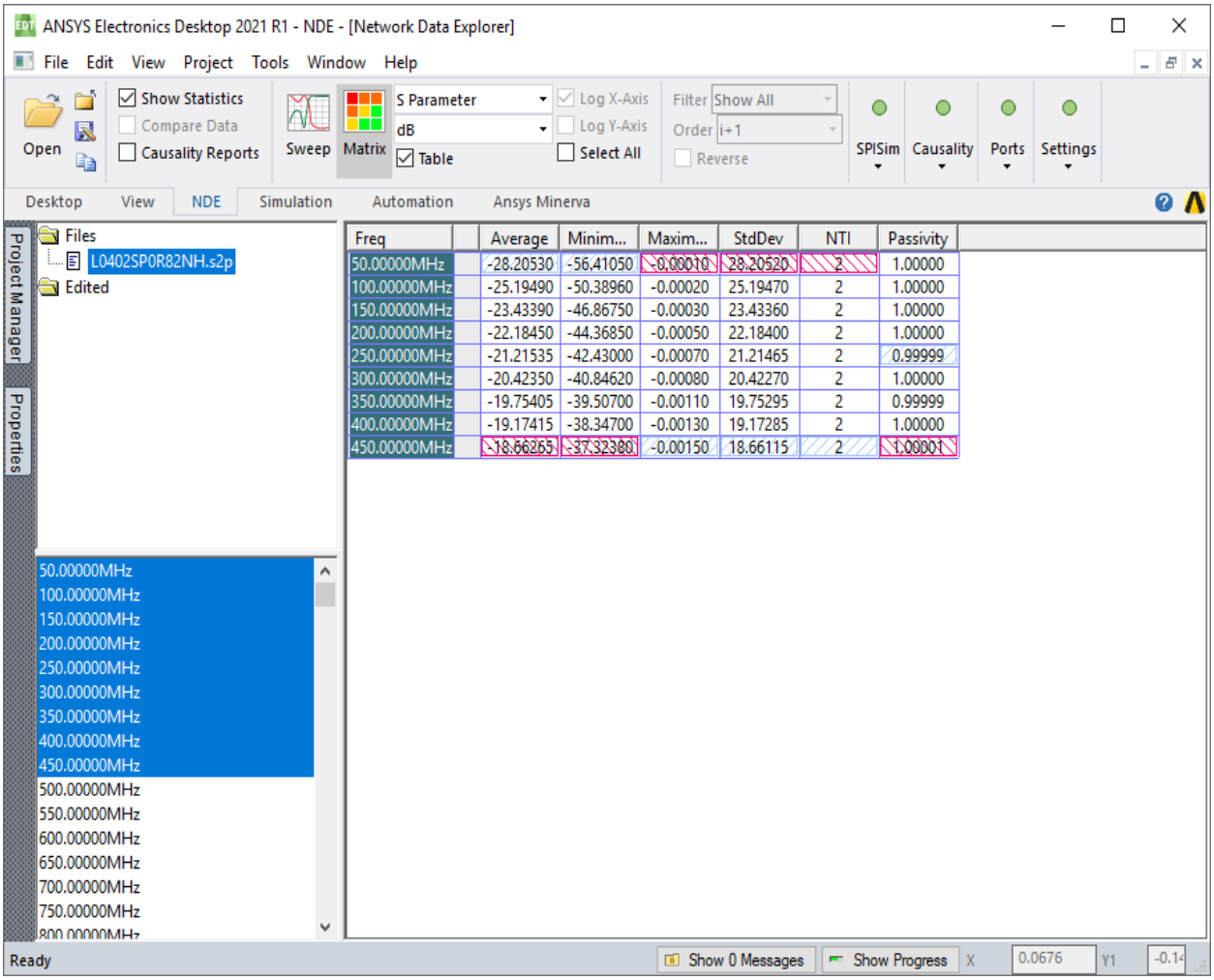
5. To add a log scale to the Y-axis, click **Log Y-Axis** on the **NDE** ribbon.
6. Right click to see a right click menu with other commands.

Displaying Matrix Statistics by Frequency

Network Data Explorer can display various statistical measurements.

1. On the **NDE** ribbon:
    1. Click **Matrix**.
    2. Click **Table**.
    3. Click **Show Statistics**.

2. In the Cell and Frequency Selection pane, select frequencies to display.



Click a column header to sort data by that column.

Hover the cursor over a cell to view information about it.

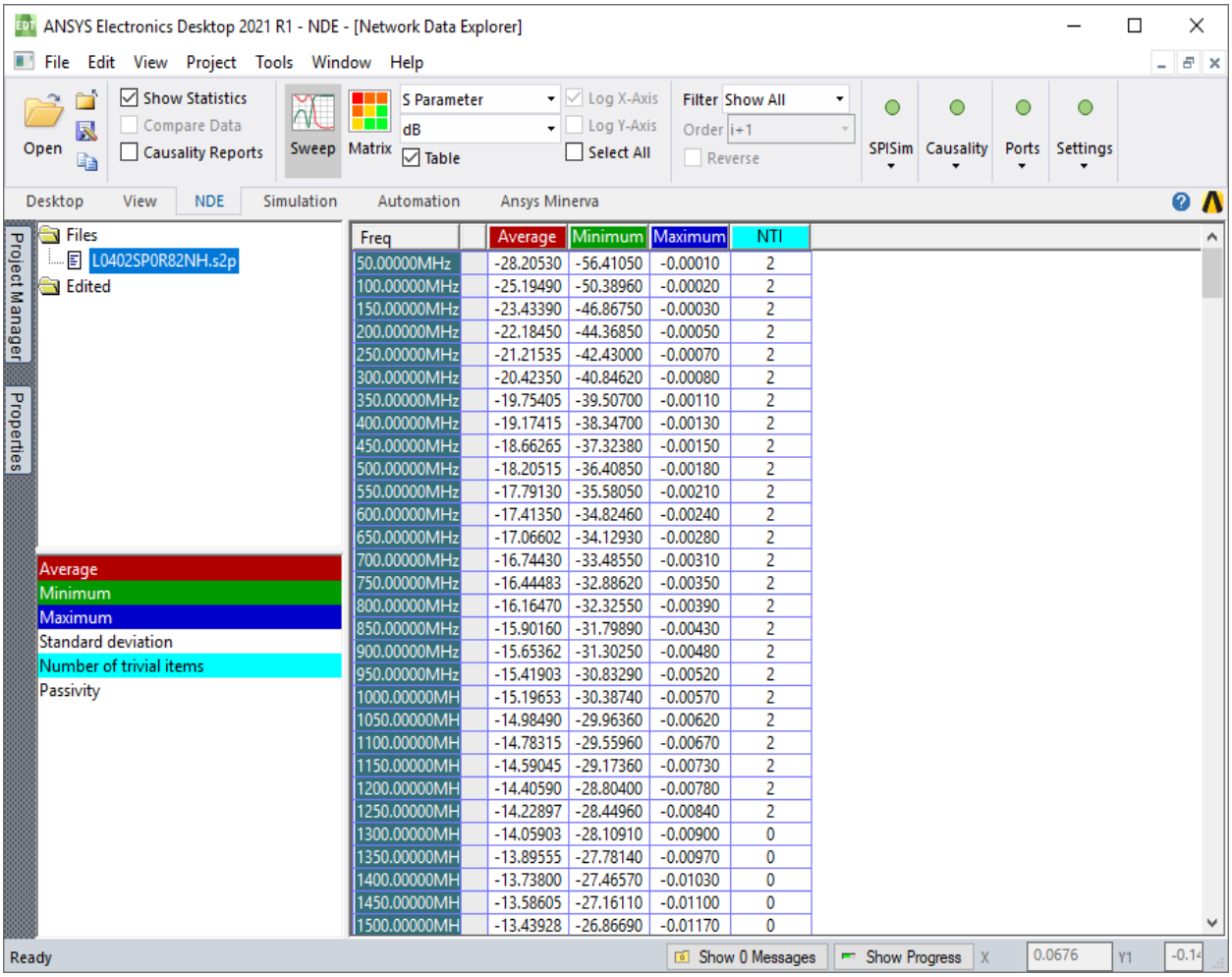
Only real (not complex) data formats are offered for statistical analysis. **Passivity** is only available for S-parameter data (comparisons inactive). **NTI** refers to the number of trivial items; for S-parameters, this includes all zeros and ones; for all other data (and data comparisons), only zeros are counted as trivial. The minimum value for each column is highlighted in blue; the maximum is highlighted in red.

Displaying Individual Statistics for All Frequencies

1. On the **NDE** ribbon:
    1. Click **Sweep**.
    2. Click **Table**.
    3. Click **Show Statistics**.

2. In the Cell and Frequency Selection pane, select statistics to display.

   The information displays in a table in the Data View pane.



Selected statistics are displayed for all frequencies.

Passivity is only available for S-parameters (comparisons inactive).

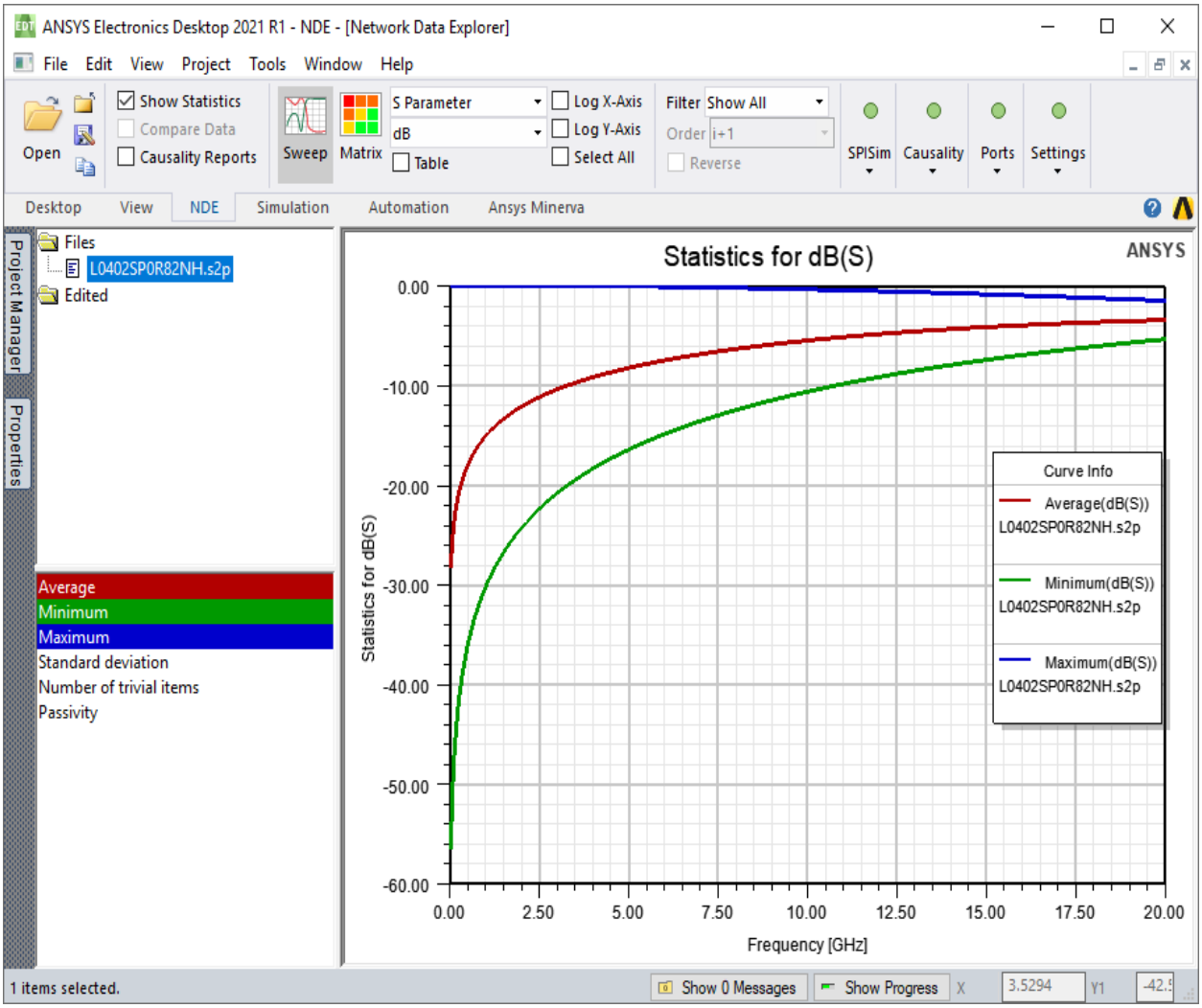
For S-parameters, the **Number of Trivial Items (NTI)** includes all values of 0 and 1. For other data and data comparisons, only values of 0 are counted as trivial.

Creating a Statistics Plot

Network Data Explorer can display a graph of selected statistical measures across all frequencies.

1. On the **NDE** ribbon:
   1. Click **Sweep**.
   2. Click **Show Statistics**.
2. In the Cell and Frequency Selection pane, select statistics to display.

The selected statistics are plotted.



Hover the cursor over a statistic to view more information about it.

Passivity is only available for S-parameters (comparisons inactive).

For S-parameters, the **Number of Trivial Items (NTI)** includes all values of 0 and 1. For other data and data comparisons, only values of 0 are counted as trivial.

Comparing Network Data

Network Data Explorer can compare variations for two network data sets that are the same size.

1. On the **NDE** ribbon, click **Sweep**.
2. In the Network Data Selection pane, select exactly two data sets.
3. On the **NDE** ribbon, click **Compare Data**.

4. In the Cell and Frequency Selection pane, select which cells you want to compare . For each value along the X-axis, the Y-axis values are subtracted, one from the other, to create the comparison plot. The second selected data is subtracted from the first selected data.
5. Optionally, check **Show Statistics** to show values applied to all cells and all frequencies

It is not possible to compare a data set against itself *unless the data set has been loaded twice*.

Traces for a given cell or statistical measure are displayed for all data sets; you can use tool tips to distinguish between them.

If a single cell or statistical measure is displayed, different colors are used for each data trace. If multiple cells or statistical measures are selected, a single color is used for all data traces for each cell or statistical measure.

In a data comparison, traces are shown for all selected data sets. This is true for both cell and statistical traces.

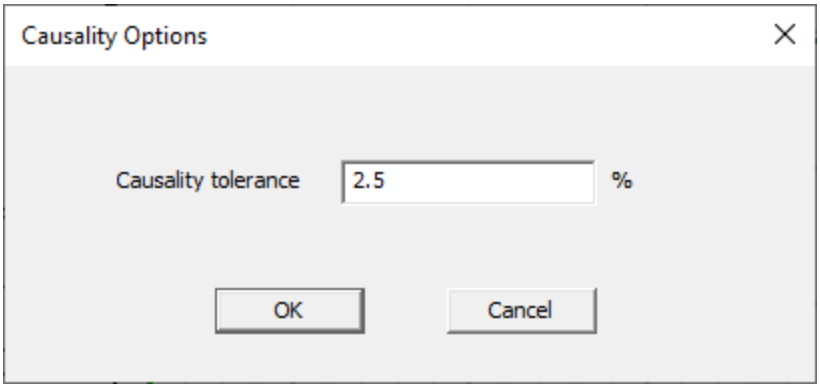
If **dB** format is shown, you can either subtract values before or after applying the dB function. See Display Format to make this choice.

Causality Checking and Plots

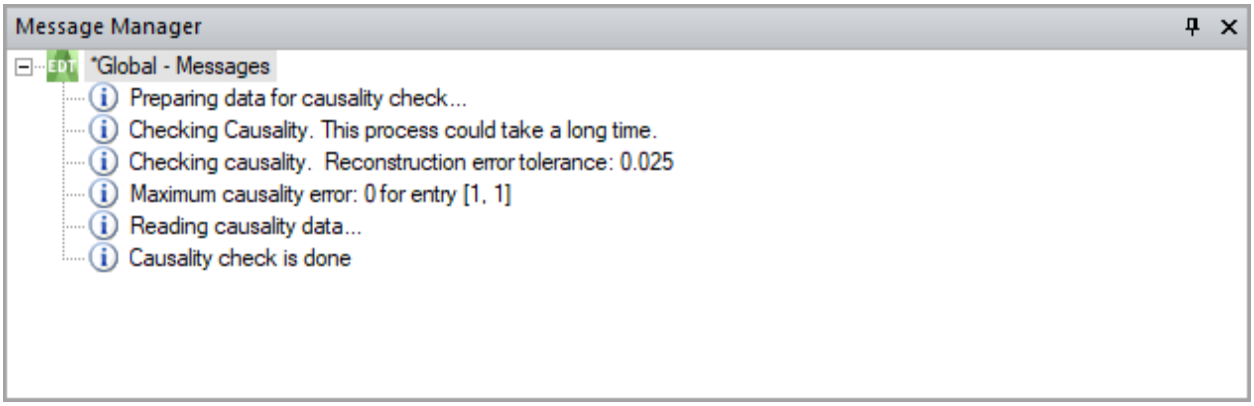
Network Data Explorer can perform a causality check on S-parameter data from any source (solution or file), and provide plots of the results in various formats.

When S-parameter data is loaded into Network Data Explorer, the **Causality** button is enabled.

1. On the **NDE** ribbon, click **Causality**. The **Causality Options** dialog opens.



2. Enter a **Causality tolerance** and click **OK** to start the causality check. Depending on the size of the S-parameter data, the causality check may take several minutes to complete. The check's status displays in the Message Manager.



When the check completes, the Messages Pane updates to display a summary of results.

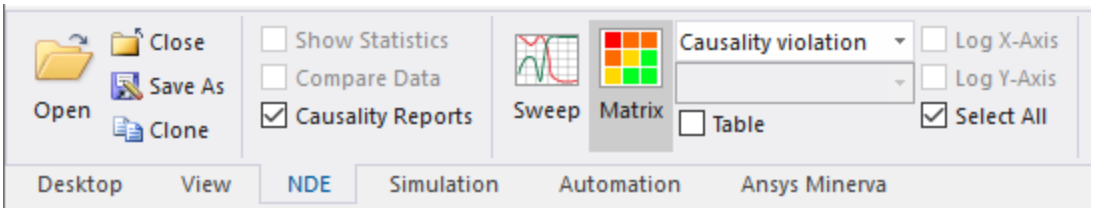
Reconstruction Error Tolerance – Causality of a frequency response is determined by calculating the generalized Hilbert transform of the data at all frequencies. A causal frequency response is equal to its generalized Hilbert transform. The reconstruction error is the difference between the tabulated data and its transform at a given frequency. The message shows the maximum reconstruction error tolerance for a causal frequency response. The default tolerance of 0.01 is equal to the state-space fitting tolerance.

Maximum Causality Error – The maximum causality error for all port pairs and all frequencies, along with the matrix indices (port numbers) where the maximum noncausality occurs. A noncausal response is one where all matrix entries can be conclusively analyzed, and at least one entry exceeds the causality tolerance. The maximum reconstruction appears first, followed by port numbers in brackets (e.g., [port number, port number]). When all results are conclusive but no matrix reconstruction error exceeds the tolerance, the maximum causality error is reported as zero, and no matrix entry is listed.

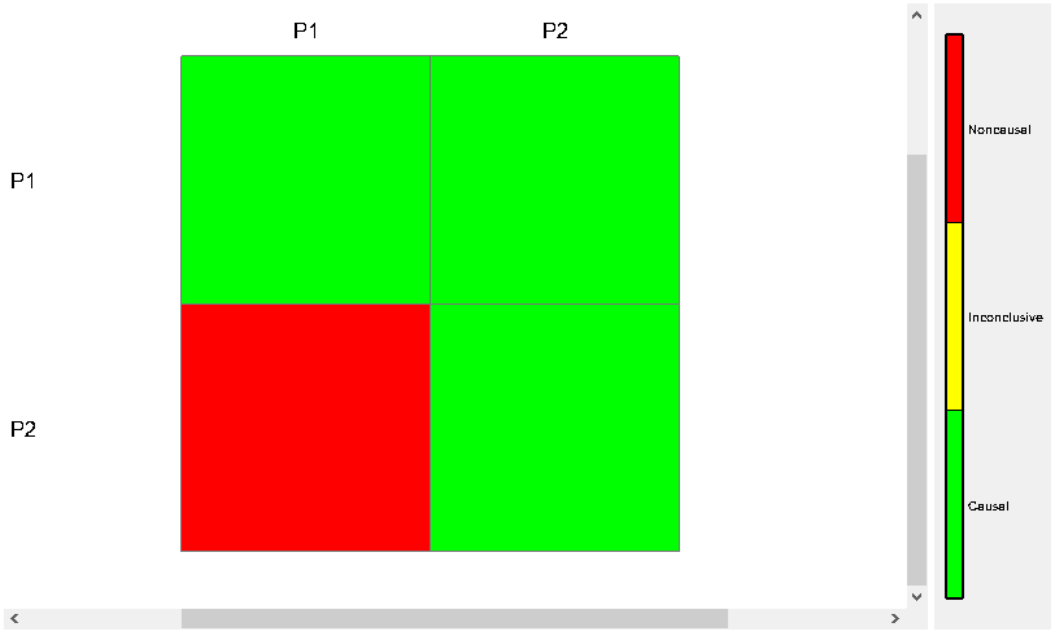
If the data does not contain enough frequency points to determine whether the data is or is not causal, the Messages Pane will note an inconclusive result. ndExplorer will also report the data set as inconclusive if any cells are inconclusive, even if other entries exhibit causality violations.

To see the plotted results of the causality check:

1. On the **NDE** ribbon, click **Causality Reports**.
2. Either click the **Matrix** icon or select **Causality violation** in the **Parameter** type drop-down menu.



A rectangular plot displays, with dimension N x N and color-coding to indicate the causality status of each port pairing.



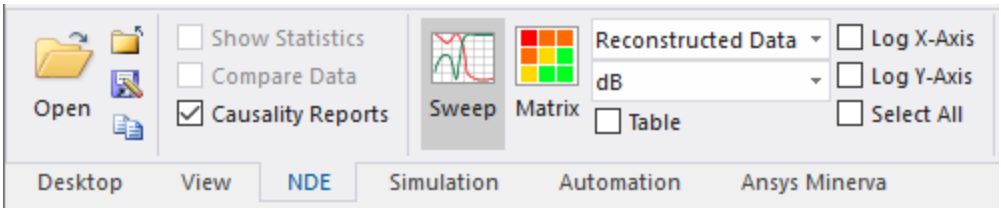
In this plot, the cells go from (Port 1, Port 1) at the upper-left area to (Port N, Port N) at the lower-right corner. The result shows the causality over all frequencies in the data. In this example, the matrix is asymmetric, so that S12 is noncausal, while S11, S21, and S22 are causal.

To see the details for each frequency, click the **Table** check button on the **NDE** ribbon.

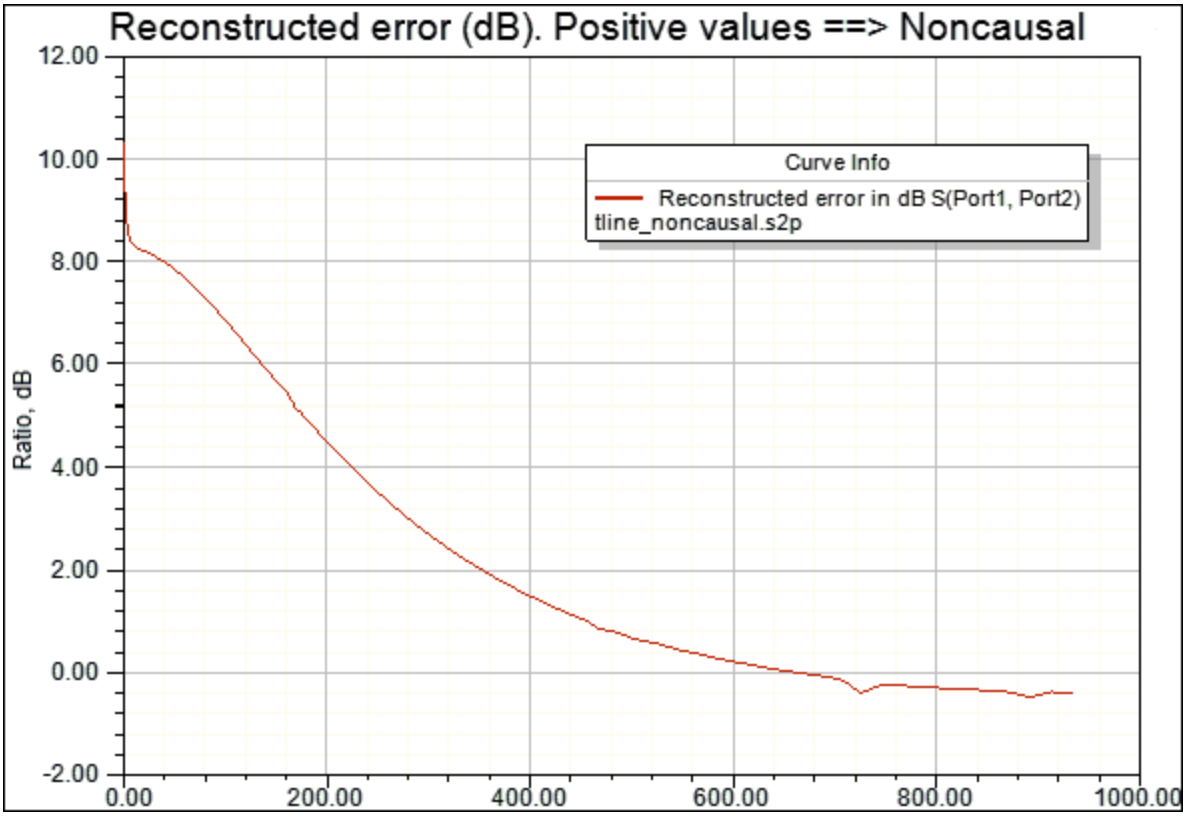
Freq		Port1	Port2	Port1	Port2
0.10000GHz	Port1	0.01715, 0.00000	0.00050, 0.00000	0.01493, 0.00000	0.00098, 0.00000
	Port2	0.53111, 0.00000	0.00579, 0.00000	0.11471, 0.00000	0.00832, 0.00000
0.20000GHz	Port1	0.01367, 0.00000	0.00059, 0.00000	0.00736, 0.00000	0.00052, 0.00000
	Port2	0.27237, 0.00000	0.00394, 0.00000	0.05933, 0.00000	0.00423, 0.00000
0.30000GHz	Port1	0.01362, 0.00000	0.00105, 0.00000	0.00344, 0.00000	0.00028, 0.00000
	Port2	0.24509, 0.00000	0.00239, 0.00000	0.01921, 0.00000	0.00195, 0.00000
0.40000GHz	Port1	0.00907, 0.00000	0.00103, 0.00000	0.00171, 0.00000	0.00006, 0.00000
	Port2	0.16665, 0.00000	0.00163, 0.00000	0.01530, 0.00000	0.00084, 0.00000
0.50000GHz	Port1	0.00693, 0.00000	0.00062, 0.00000	0.00075, 0.00000	0.00006, 0.00000
	Port2	0.11877, 0.00000	0.00106, 0.00000	0.00542, 0.00000	0.00040, 0.00000
0.60000GHz	Port1	0.00295, 0.00000	0.00044, 0.00000	0.00047, 0.00000	0.00005, 0.00000
	Port2	0.05535, 0.00000	0.00042, 0.00000	0.00462, 0.00000	0.00015, 0.00000
0.70000GHz	Port1	0.00015, 0.00000	0.00009, 0.00000	0.00029, 0.00000	0.00003, 0.00000
	Port2	0.00510, 0.00000	0.00012, 0.00000	0.00114, 0.00000	0.00008, 0.00000
0.80000GHz	Port1	0.00058, 0.00000	0.00068, 0.00000	0.00047, 0.00000	0.00004, 0.00000
	Port2	0.02772, 0.00000	0.00088, 0.00000	0.00115, 0.00000	0.00010, 0.00000
0.90000GHz	Port1	0.00210, 0.00000	0.00081, 0.00000	0.00040, 0.00000	0.00013, 0.00000
	Port2	0.05492, 0.00000	0.00133, 0.00000	0.00173, 0.00000	0.00010, 0.00000
1.00000GHz	Port1	0.00221, 0.00000	0.00064, 0.00000	0.00033, 0.00000	0.00010, 0.00000
	Port2	0.07703, 0.00000	0.00142, 0.00000	0.00111, 0.00000	0.00009, 0.00000
1.10000GHz	Port1	0.00410, 0.00000	0.00045, 0.00000	0.00037, 0.00000	0.00008, 0.00000
	Port2	0.08851, 0.00000	0.00151, 0.00000	0.00137, 0.00000	0.00011, 0.00000
1.20000GHz	Port1	0.00433, 0.00000	0.00078, 0.00000	0.00017, 0.00000	0.00013, 0.00000
	Port2	0.09013, 0.00000	0.00100, 0.00000	0.00168, 0.00000	0.00017, 0.00000
1.30000GHz	Port1	0.00377, 0.00000	0.00072, 0.00000	0.00020, 0.00000	0.00018, 0.00000
	Port2	0.09487, 0.00000	0.00151, 0.00000	0.00081, 0.00000	0.00012, 0.00000
1.40000GHz	Port1	0.00380, 0.00000	0.00035, 0.00000	0.00020, 0.00000	0.00014, 0.00000
	Port2	0.08918, 0.00000	0.00136, 0.00000	0.00139, 0.00000	0.00009, 0.00000
1.50000GHz	Port1	0.00417, 0.00000	0.00005, 0.00000	0.00012, 0.00000	0.00011, 0.00000
	Port2	0.08091, 0.00000	0.00174, 0.00000	0.00094, 0.00000	0.00021, 0.00000
1.60000GHz	Port1	0.00315, 0.00000	0.00038, 0.00000	0.00006, 0.00000	0.00015, 0.00000
	Port2	0.07512, 0.00000	0.00208, 0.00000	0.00065, 0.00000	0.00011, 0.00000
1.70000GHz	Port1	0.00275, 0.00000	0.00034, 0.00000	0.00006, 0.00000	0.00021, 0.00000
	Port2	0.06069, 0.00000	0.00152, 0.00000	0.00101, 0.00000	0.00021, 0.00000

To plot the reconstructed frequency response generated by the causality checker:

1. On the **NDE** ribbon, click **Causality Reports**.
2. Either click the **Sweep** icon or select **Reconstructed Data** in the **Parameter** type drop-down menu.
3. In the **Format** drop-down menu, select the desired format. **dB** is selected by default.

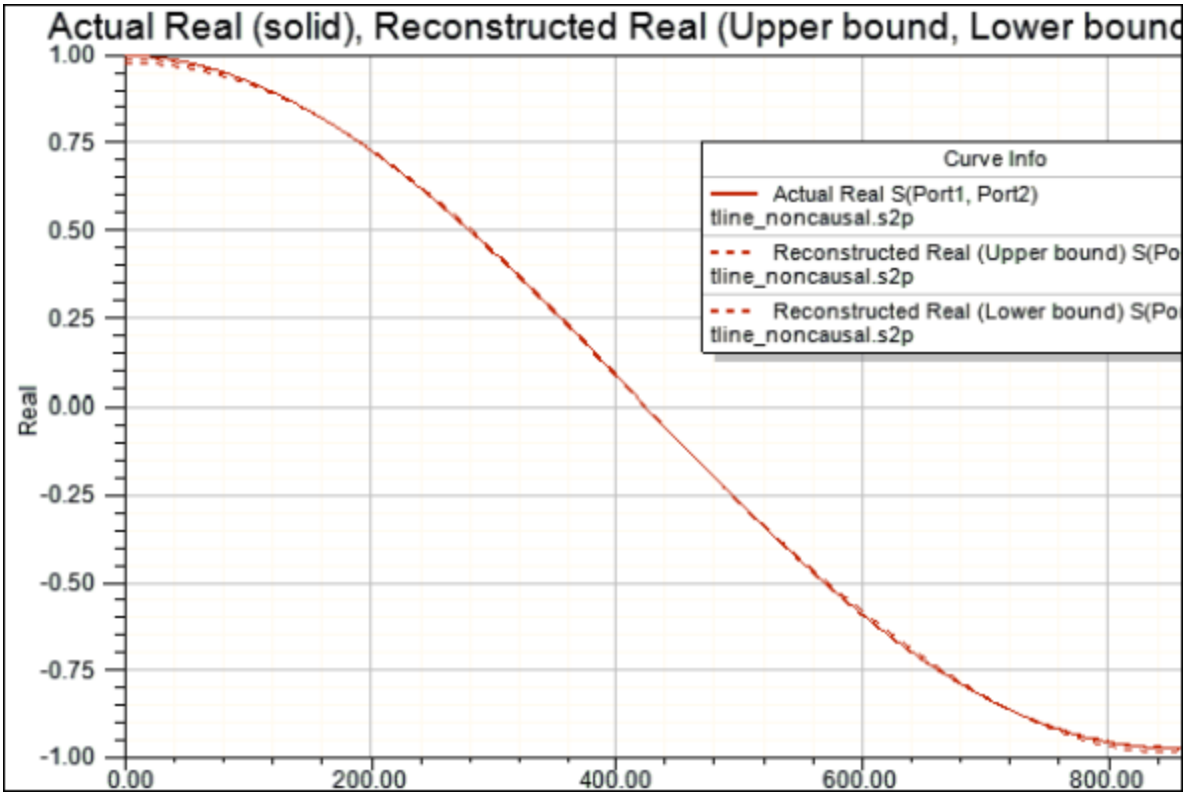


The plot appears, showing the reconstruction error at each frequency divided by the tolerance.



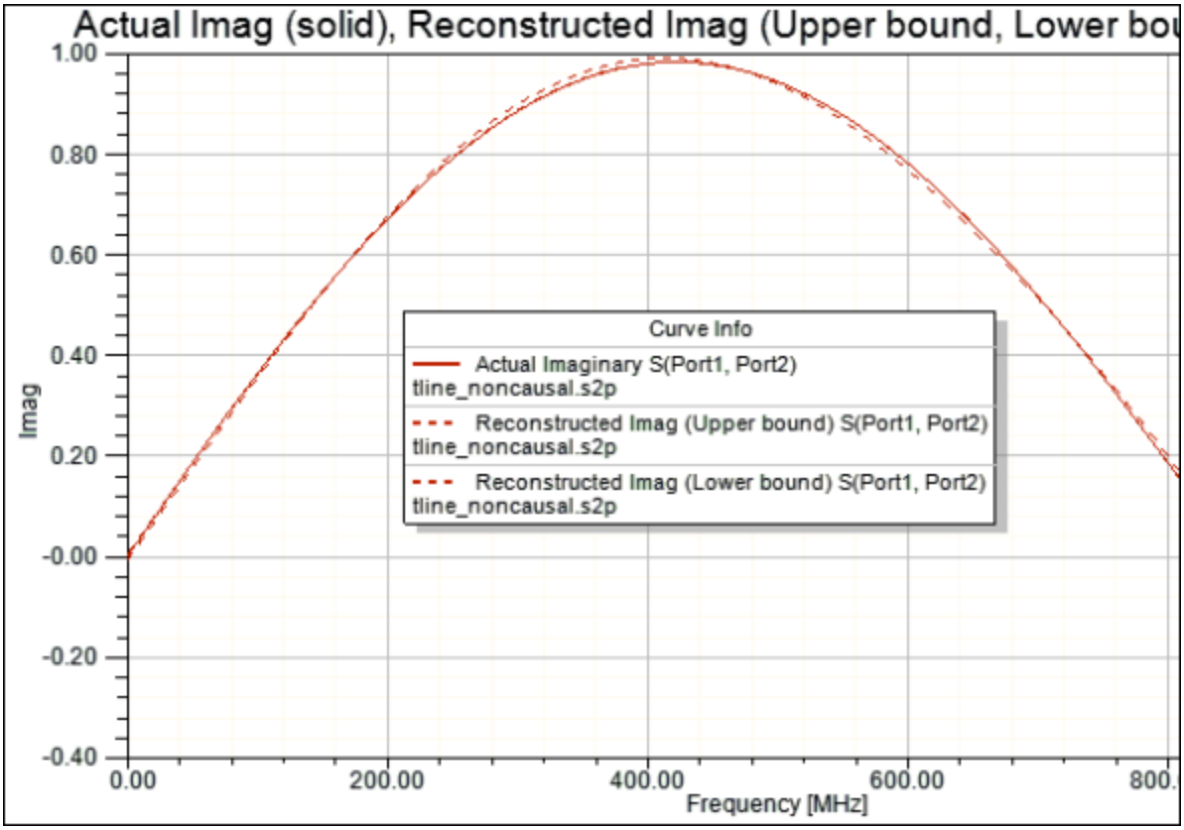
The reconstruction error ratio for parameter S12 is positive for frequencies less than about 680MHz, indicating a broad range of noncausal behavior.

4. To compare the real part of the reconstructed data to the real part of the actual data, set the **Format** to **Real**.



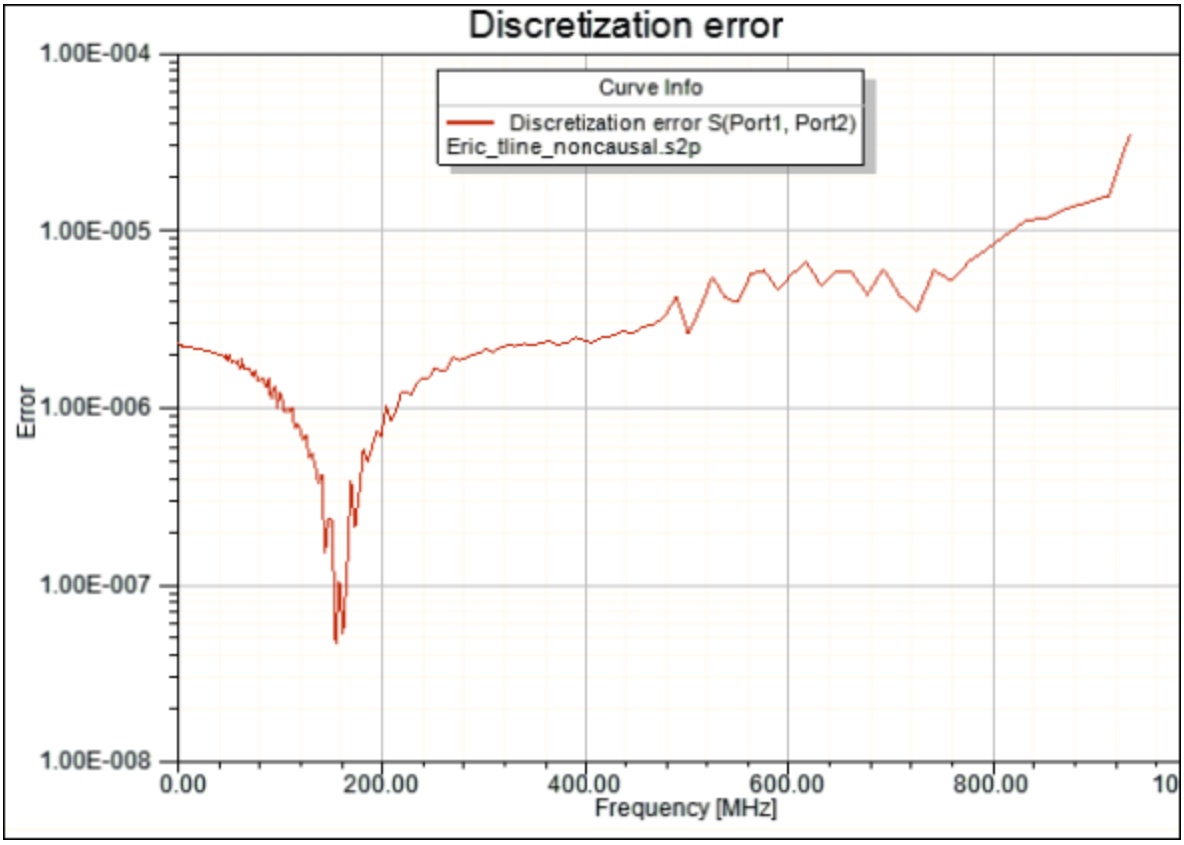
For a causal frequency response, the actual data (solid line) will be within the upper and lower bounds of the reconstruction (dotted lines) at all frequencies.

5. To compare the imaginary part of the reconstructed data to the imaginary part of the actual data, set the **Format** to **Imaginary**.



For a causal frequency response, the actual data (solid line) will be within the upper and lower bounds of the reconstruction (dotted lines) at all frequencies.

6. To view the frequency-dependent discretization error, set the **Format** to **Discretization**.



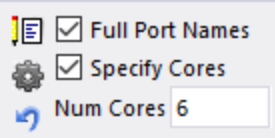
The discretization error is the error that is due to the fact that the data are available only at discrete frequencies rather than for a continuous spectrum. A discretization error near or greater than the causality tolerance renders the causality check inconclusive. Data at more frequencies could reduce the discretization error and render the analysis conclusive. This set of data exhibits low discretization errors (<<0.01) at all frequencies, and the causality check is conclusive (conclusively noncausal in this example).

Network Data Explorer Multithreading

By default, multithreading (execution on multiple cores) is enabled for Causality Checking and Macro Model export. Multithreading saves significant time in the Causality Check calculation, and improves the time for other state-space fitting operations. See Technical Notes.

The **Cores** field in the Network Data Explorer Control Pane defaults to half the number of cores detected on your computer.

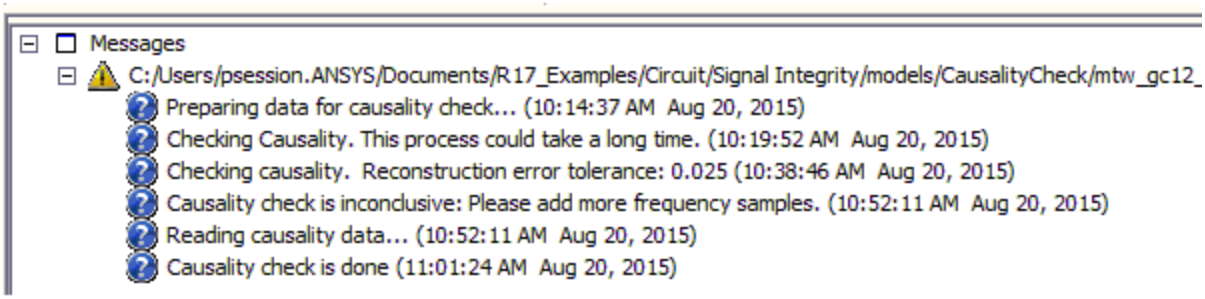
For best performance, disable hyper-threading on your computer. When hyper-threading is enabled, the number of cores includes the physical cores and an equal number of logical cores. With hyper-threading disabled, the display shows only half the number of physical cores.



To disable multithreading, uncheck the checkbox.

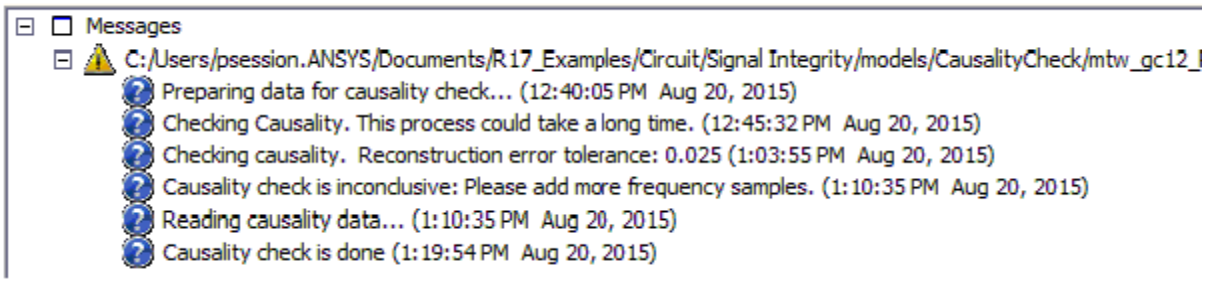
Multithreading Technical Notes

Multithreading can save significant time in the causality check calculation, especially for data sets with large number of ports or a large number of frequencies. The Network Data Explorer contributes a fixed amount of overhead time in preparing the data, and this overhead is not reduced by multithreading. Here are the messages from the causality check of a 278-port Touchstone file using two cores.



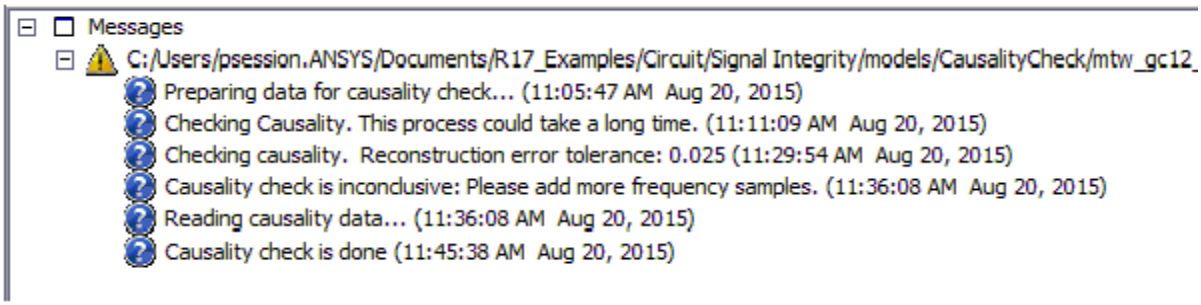
The causality check itself happens between the message `Checking causality. Reconstruction error tolerance: 0.025` (time 10:38:46) and the message `Causality check is inconclusive: Please add more frequency samples` (time 10:52:11). The difference between these two times is the time for the causality check after setup, 13:25 or thirteen minutes and twenty-five seconds. There is a fixed overhead of around 25 minutes involved in the overall time duration for this example.

Here are the causality check messages for the same 278-port file using 8 cores:



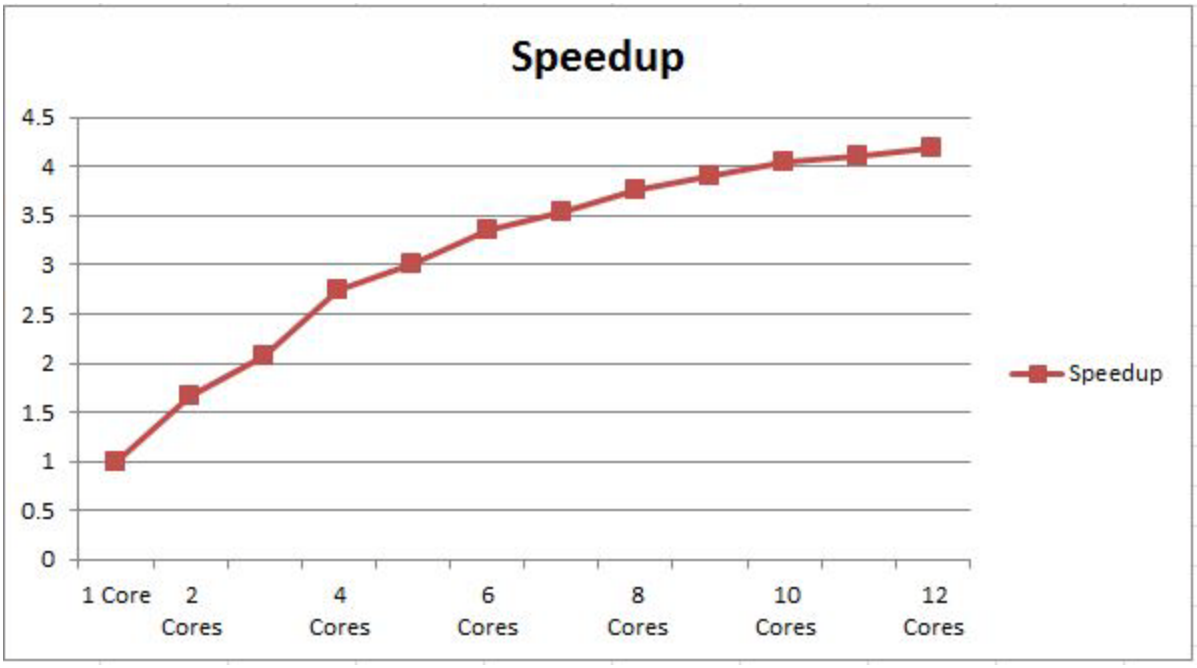
Now the causality check happens between times (1:03:55) and (1:10:35), a difference of (6:40) or six minutes and forty seconds, about half the time for the two-core example.

The speedup from multithreading is not linear in practice. Here are the messages from the same 278-port file using 16 cores:



Now the causality check happens between times (11:29:54) and (11:36:08), a difference of (6:14) or six minutes and fourteen seconds, not significantly better than the eight-core performance.

Here is a graph showing the speedup of the causality checker using multithreading on this 278-port file. The plot is generated using data from runs with 1, 2, 4, 6, 8, 10, and 12 cores, then averaging the known times to approximate the speedup for 3, 5, 7, 9, and 11 cores. The speedup for N cores is the time with one core divided by the time with N cores.



23 - Post Processing and Generating Reports

When Maxwell has completed a solution, you can display and analyze the results in the following ways:

- View solution data including the following: convergence information, computing resources that were used during the solution process, mesh statistics, and matrices computed during each adaptive, non-adaptive, or sweep solution.
- View analysis results for Optimetrics solutions.
- Plot field overlays - representations of basic or derived field quantities on surfaces or objects.
- Overlay Field Plots on Models
- Create 2D or 3D reports of RLC matrices and basic and derived field quantities.
- Plot the finite element mesh on surfaces or within 3D objects.
- Create animations of field quantities, the finite element mesh, and defined project variables.
- Use the Fields Calculator to perform computations using basic field quantities.

Note	Except in the case of non-model boxes drawn in the global coordinate system (CS), non-model objects cannot be used for any fields post processing operation You can use non-model boxes drawn in the global CS for post processing operations, including integration and solution domaining.

Related Topics

Viewing Analysis Results for Optimetrics Solutions

Viewing Solution Data

While Maxwell is generating a solution, or when it is complete, you can view the following types of information about the solution in the **Solutions** dialog box. The information actually available will vary with the solution type and setup of your design.

- Convergence information.

Note	Since adaptive mesh refinement is not performed for transient solutions, convergence data is not available for **Transient** analysis.

- Computing resources, or profile information, that were used during the solution process.
- Output Parameters (for example Force and Torque), and Matrix Data computed during each adaptive, non-adaptive, or sweep solution.
- Mesh data.
- The state of solved solutions.
- Winding results (for eddy current solutions only).
- End Connection results (for eddy current solutions only).
- Loss results (for eddy current solutions only).

To access the **Solutions** dialog box, in which the information above can be viewed, do one of the following:

- Click **Maxwell3D**, **Maxwell2D,** or **RMxprt**, and then select **Results>Solution Data**  .
- Right-click **Results** in the project tree, and then click **Solution Data** on the shortcut menu.

Related Topics

Viewing Solution Data for an Optimetrics Design Variation

Monitoring the Solution Process

Viewing Convergence Data

To view an adaptive solution's convergence information, either during or after the solution process:

1. In the project tree, right-click the solution setup of interest, and then click **Convergence** on the shortcut menu.

   The **Solutions** dialog box appears with the **Convergence** tab selected.
2. From the **Simulation** pull-down list, select the solution setup for which you want to view convergence data.

   By default, the most recently solved solution is selected.
3. Under the **Convergence** tab, depending on your design setup, you can review the following convergence data:
   - Whether the solution is converged or not converged.
   - Number of adaptive passes completed and remaining.
   - The number of tetrahedra or triangles created at each adaptive pass.
   - Solution type dependent parameters such as Loss, Total Loss, Loss Error.
4. Select **Table** to display the convergence data in table format or **Plot** to plot the convergence data on a rectangular (XY) plot.
5. Optionally you can **Export** the convergence data to a text file.
6. If you select **Save Defaults**, the current selection of either Table or Plot, and for Plot the current X and Y axis selections, will be applied based on the design-type, each time you open the desktop. You can also select **Clear Defaults** to reset to the original settings.

> **Note:** In the case of the 3D AC Conduction solver, even though the convergence columns names refer to the loss as a solution criteria, the convergence parameter considered is the magnitude of the apparent power.

Related Topics

Viewing Solution Data for an Optimetrics Design Variation
Defining Expressions for Non-Transient Solutions
Plotting a Quantity versus Adaptive Pass

Viewing the Number of Completed Passes

At any time during the solution process, you can view the number of adaptive passes (solve — error analysis — refine cycles) that have been completed as well as the maximum and minimum number of passes. When the solution is complete, you can view the number of adaptive passes that were performed. If the solution converged within the specified stopping criteria, fewer passes than requested may have been performed.

To view the number of passes:

- In the project tree, right-click the solution setup of interest, and then click **Convergence** on the shortcut menu.

  The **Solutions** dialog box appears with the **Convergence** tab selected.

  The number of completed passes is listed in the **Number of Passes** area.

Viewing Output Variable Convergence

At any time during or after the solution process, you can view the values of the output variable.

To view the output variable convergence, use the Reporter to create a plot that displays the output variable values.

Related Topics

Specifying Output Variables
Defining Expressions for Non-Transient Solutions

Plotting Convergence Data

To display convergence data vs. pass on a rectangular (x - y) plot:

1. In the project tree, right-click the solution setup of interest, and then click **Convergence** on the shortcut menu.

   The **Solutions** dialog box appears with the **Convergence** tab selected.
2. In the lower-left corner of the window, select **Plot** as the **View** type.
3. Select the data you want to plot on the x-axis from the **X** pull-down list.
4. Select the data type you want to plot on the y-axis from the **Y** pull-down list.
5. Click **Close**.

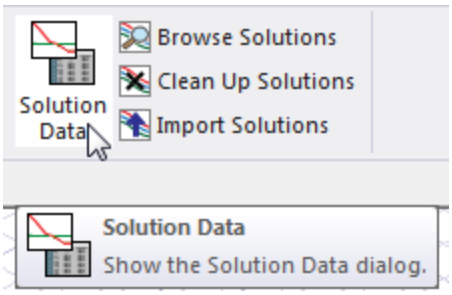
The xy plot appears in the view window.

Viewing a Solution Profile

At any time during or after the solution process, you can examine the computing resources or profile data that were used by Ansys Electronics Desktop solvers during the analysis. The profile data is essentially a log of the tasks performed by Ansys Electronics Desktop during analysis. The log indicates the length of time each task took and how much physical and disk memory was required.

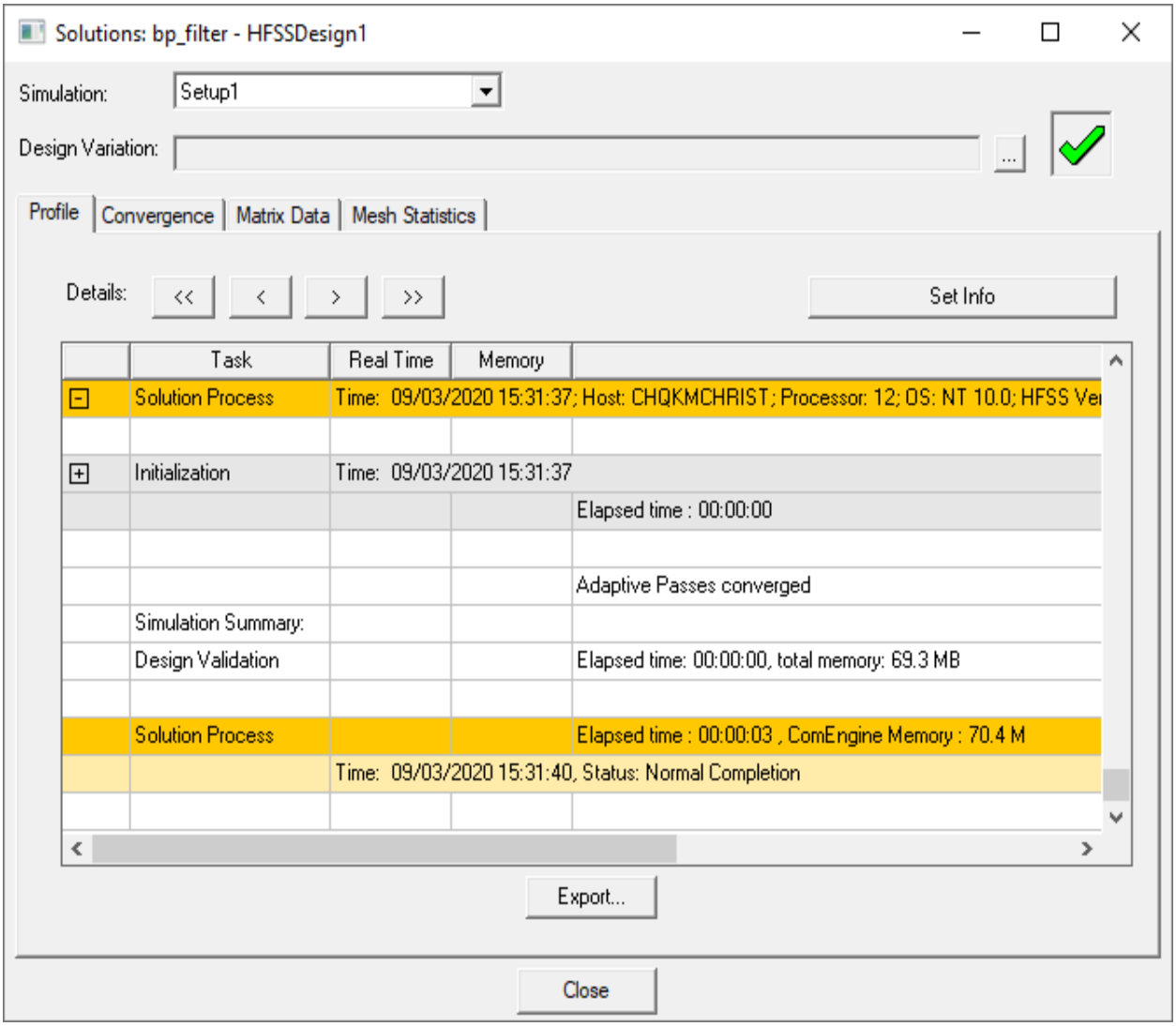
1. In the project tree, right-click a solution setup, and select **Profile**.

   The **Solutions** dialog box appears, with the **Profile** tab selected. You can also click the **Solution Data** icon on the ribbon to open the **Solutions** dialog box, and then select the **Profile** tab.

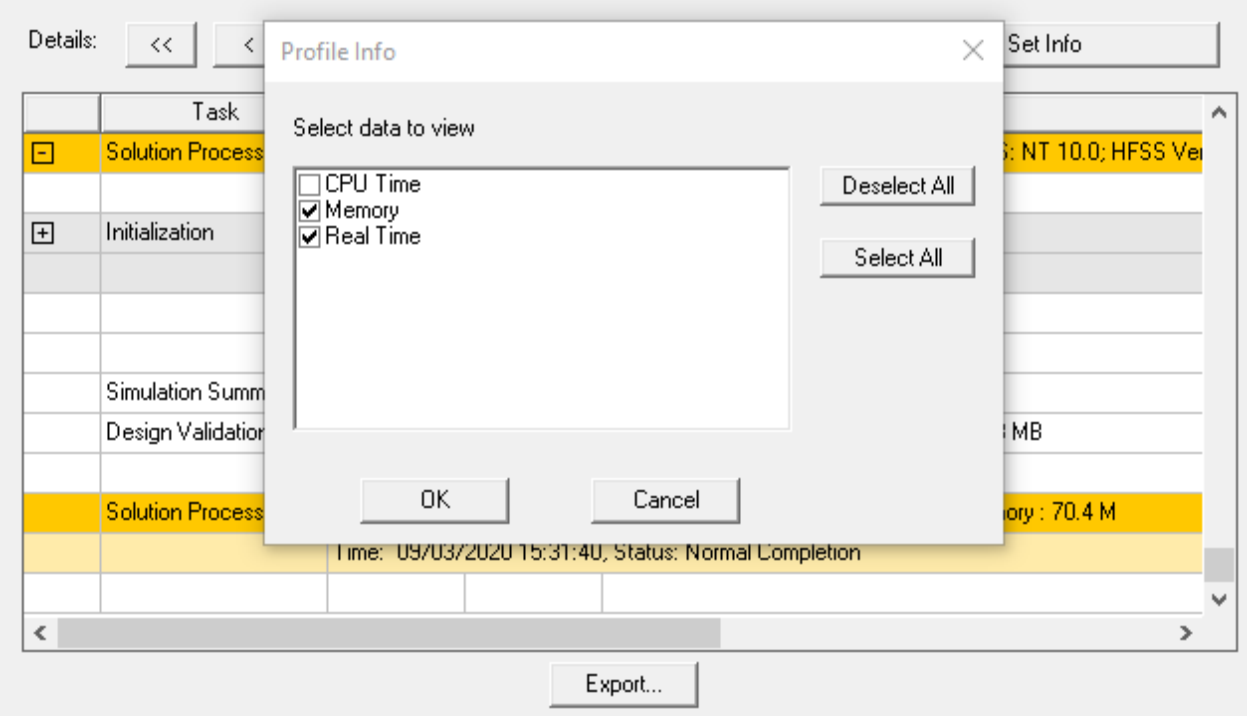


   The displayed data depends on the type of problem and solution setup.

   A Beta feature (Enhanced Profile Display] allows you to have the **Profile** data is displayed using nested subgroups. The **Details** buttons compress groups [<<] and [<] and expand them [>] and [>>].



The Profile data is displayed using nested subgroups. The **Details** buttons let you compress groups [<<] and [<] and expand them [>] and [>>]. You can also click the [+] buttons in the first column to open a subgroup, or use a [-] button to close one. The Set Info button lets you select from available Profile information.



2. If one or more setups exist, you can select the desired profile information from the **Simulation** drop-down menu. In general, the information displayed includes:

Task	Lists the type of task that was performed. The Tasks lists included Start, various Mesh tasks, Simulation Setup, Port Adaptation, Adaptive Pass tasks, including simulation setup, Matrix Assembly, Solver tasks, and Field Recovery, Sweep tasks, and Solution Process summary and Totals for time.
Real Time	The difference in time between the start of the task and the end of the task (elapsed time).
CPU Time	The amount of CPU time required to perform the task.
Memory	The peak amount of physical memory (RAM) used by the individual executable running the task. The memory is freed for other uses after each task is complete.
Information	General information about the solution, for example, the number of tetrahedra used in the mesh, disk use, solver information, sweep information, the elapsed time, and the memory used by the MaxwellComEngine, and totals. The elapsed time is the difference between the time in the **Start** line and the time in the **Total** line — how long the task actually took between when **Analyze** was clicked and when the simulation finished. The elapsed time is a little greater than the **Real Time** given in the **Total** line. This is because the **Total** line contains the sum of all of the real times reported by the

	individual processes and does *not* include the typically small amount of time the Maxwell ComEngine uses before, after, and in between calling the other executables.

The matrix solver writes specific information in some of these fields as outlined below:

Task	The matrix solver task reports the type of solution performed by the solver, based on the physics of the problem. It has the form "Solver *pdsn*" (e.g. Solver MRS2 or Solver DCS4-L2), where • *p* , the precision type is: M (mixed for direct solver) or D (double for direct and iterative solver). • *d* , the matrix data type is: R (real) or C (complex) • *s* , the symmetry type is: S (symmetric), A (asymmetric), or H (hermitian) • *n* , the number of processors used. You specify the number of available processors on the local machine in the Maxwell options. If a solve does not use all available processors (local or distributed), the number reported may be less than the number available. If a simulation uses the iterative solver, the Solver designation can include a level indicator appended. to an Iterative solver designation (L2 in the example above). The higher the Level number the lower the memory, you will never see L1 (this would be equivalent to direct solver. And a first order solve will only display L2 since it only has one level of order to go down for preconditioning. A second or mixed order solve may display L3 depending on the mesh quality. If the solver switches from the Iterative Solver to the Matrix solver, you see a Matrix solver warning: Switch from Iterative Solver to Direct Solver.
Information:	The matrix solver information line includes, for example, Disk = 0 KBytes, matrix size 11137 , matrix bandwidth 20.3) • **Disk**: The amount of hard disk space used during the calculation of the matrix solution. If the disk usage for matrix solver is non-zero in profile, it usually indicates off-core matrix solver. If the matrix solver must solve off-core, smaller blocks of the data to be solved are created on disk, each block is then solved in physical memory, and then the matrix solution is reassembled. As a result of this additional processing, the time required to calculate a solution is higher. • **Matrix size**: The size of the matrix that was solved (the number of unknowns)

	• **Matrix bandwidth**: An FEM matrix is a sparse matrix. The solver only stores the non-zero entries. The matrix bandwidth is the average number of non-zeros per row. It gives an idea of the sparsity of a FEM matrix.Storage for the sparse matrix is proportional to the total number of nonzeros = #rows x bandwidth. The higher the bases order, the larger the bandwidth. • In the case of the Iterative Solver, **# Iterations**

To Export the Profile data:

1. Open the **Solutions** dialog box with the **Profile** tab selected.
2. Click the **Export...** button.

   This opens a file save dialog that lets you provide a file name and location.
3. Click **Save**.

   The data is saved in a text file with a **.prof** extension.

Related Topics

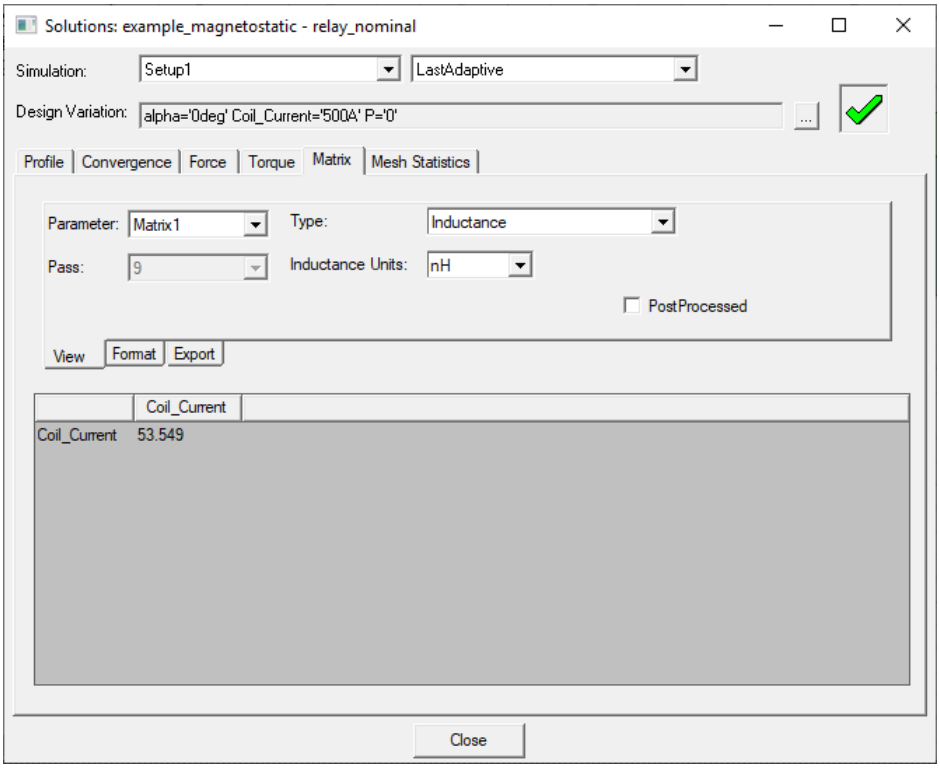
Viewing an Optimetrics Solution's Profile Data

Viewing Matrix Data

To view matrix parameter data:

1. In the project tree, right-click the solution setup of interest, and then click **Solutions** on the shortcut menu.

   The **Solutions** dialog box appears. Select the **Matrix** tab. A typical example is shown below.



2. The **Design Variation** box shows the current design.

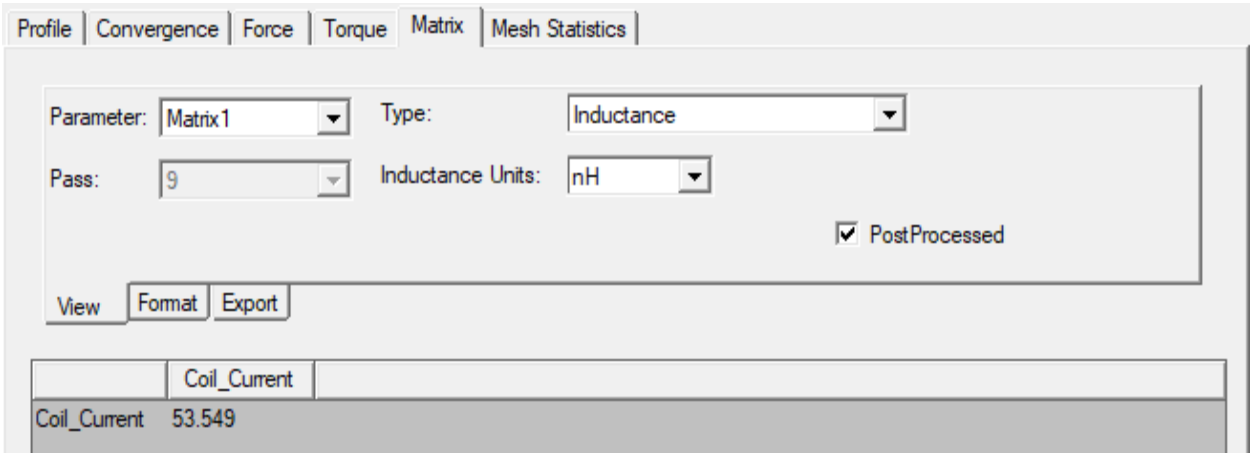
   Optionally, click the **...** button, and choose a design variation solved during an optimization or parametric analysis from the **Set Design Variation** dialog box which lists all the solved variations in the design.
3. In the **Simulation** pull-down lists:
   - Select the solution setup from the left pull-down list.
   - Select the solved pass data you wish to view from the right pull-down list, either the **LastAdaptive** pass or an intermediate **AdaptivePass**. If you select **AdaptivePass** you must also select the desired intermediate pass form the **Pass** pull-down list.
4. Select a **Parameter**.
5. Select one of the items from the **Type** pull-down list. Items available in the **Type** list depend upon the design's solution type setting. For example:
   - **Inductance** (for magnetostatic and 3D transient A-Phi solutions)
   - **Re(Z), Im(Z)** (for eddy current solutions)
   - **R, L** (for eddy current solutions. The extracted resistance values of designs that use Litz wire windings take into consideration the stranded AC loss effects due to skin effect and proximity effects.)
   - **G,C** (for ACConduction solutions)
   - **Capacitance** (for electrostatic and 3D transient A-Phi solutions)

- **Conductance** (for DC Conduction solutions)
- **Flux Linkage** (for magnetostatic and eddy current solutions)
- **Inductive Coupling Coefficient** (for magnetostatic and eddy current solutions)
- **Capacitive Coupling Coefficient** (for electrostatic and ACConduction solutions)
- **Conductive Coupling Coefficient** (for DC Conduction solutions)

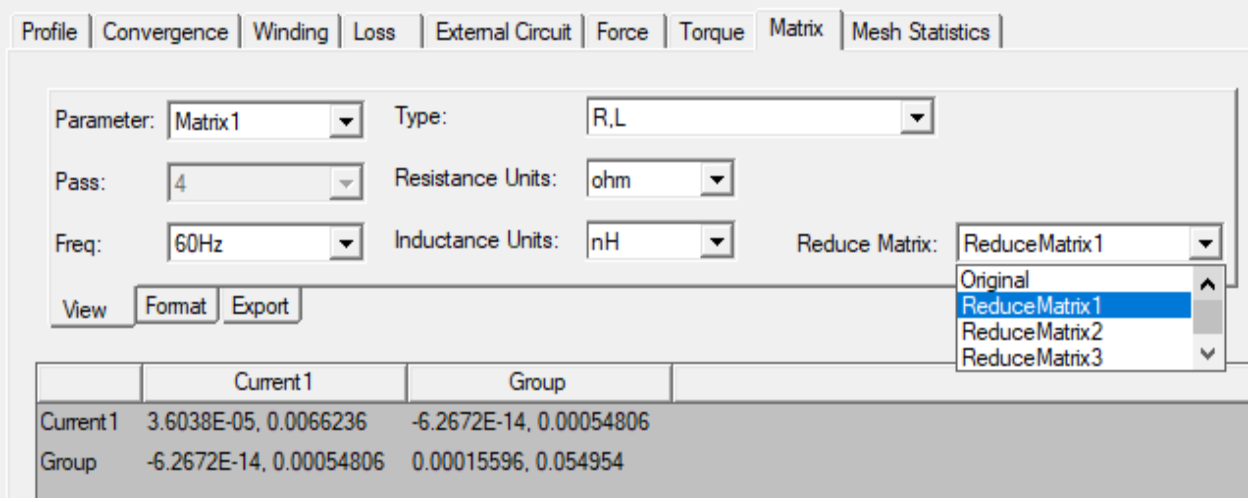
Depending on the solution type you selected, you may have to specify the **Units** in which to display the information. The available units depend on the matrix type being displayed.

The bottom pane displays values for the selected type. The **Format** tab allows you to customize how values are displayed by setting the desired width, precision, and whether to use scientific notation.

6. (Optional) For Magnetostatic designs, the results of the post processing calculation on the matrix, including grouped sources defined on the Matrix dialog Post Processing tab, can be seen by selecting **PostProcessed**.



7. (Optional) For 2D and 3D Eddy Current designs, you can join (group) two or more excitations to one excitation in either a series or a parallel connection. The result of each set of join (group) operation is known as a Reduce Matrix. For Eddy Current designs that include one or more reduce matrices, select the reduce matrix of interest from the **Reduce Matrix** drop down list. For reduce matrix selections, only the **R, L Type** values are available for viewing. (Refer to *Assigning a Reduce Matrix* for information on creating reduce matrices.)



8. (Optional) To export the matrix data (as specified by the **Type** list):
   a. On the **Export** tab, click **Export Solution**.
      The **Export Solution** dialog box appears.
   b. Select a location for the solution data.
   c. Type a filename in which to store the data in the **File name** text box.
   d. Click **Save**. The solution is saved as a data table in the specified file.
9. (Optional) For 2D and 3D Magnetostatic designs only, you may export a circuit based on the matrix parameter type by clicking **Export Circuit** on the Export tab. The **Export Circuit** button appears in the dialog box when a supported type is selected.

Related Topics

Assigning a Matrix

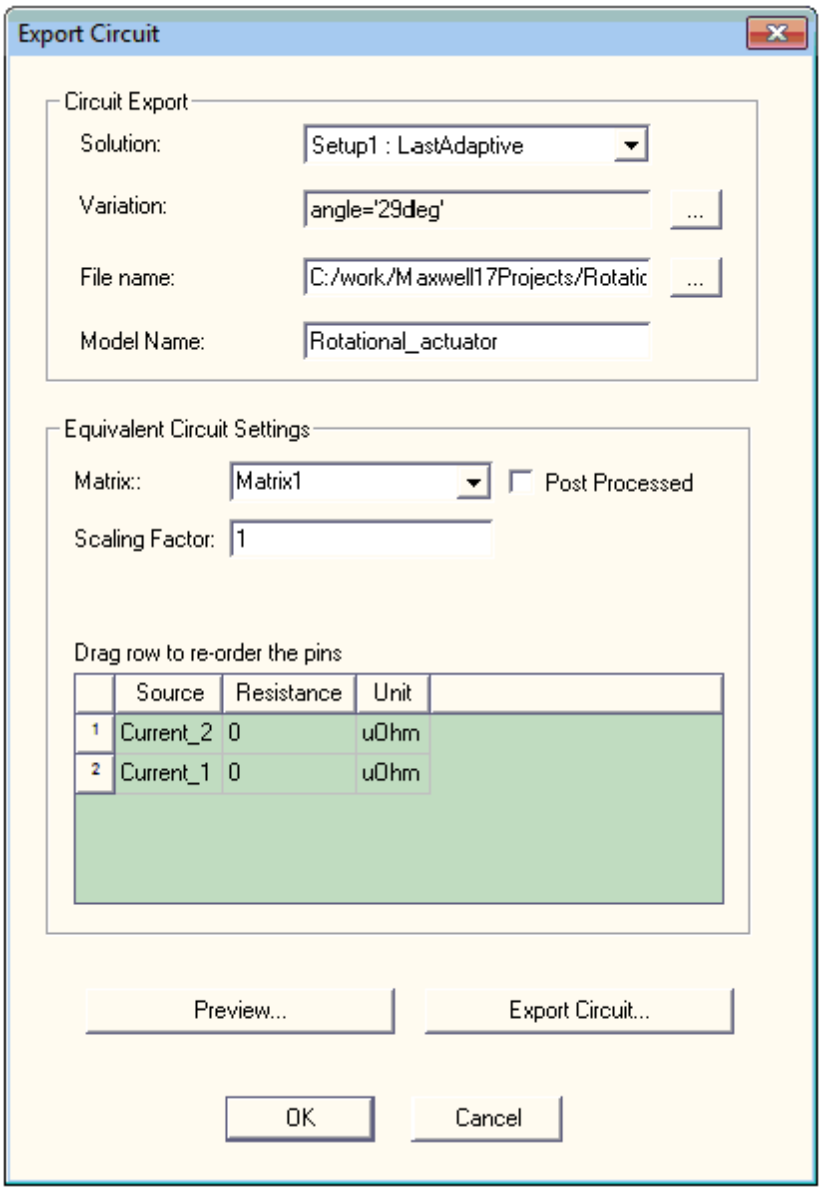
Assigning a Reduce Matrix

AC Loss Effects of Litz Wire on Impedance Matrix

Exporting a Circuit Model

To export the circuit:

1. In the **Solutions** dialog box, click **Export Circuit**.
   The **Export Circuit** dialog box appears.



2. In the **Equivalent Circuit Settings** section, select the matrix you want to export from the **Matrix** pull-down list.
3. (Optional) Type a scaling factor in the **Scaling Factor** text box. You may want to provide a scaling factor if you are using symmetry to analyze a portion of the model.
4. For an induction matrix, type a **Resistance** for each source listed in the table.
5. (Optional) If desired, drag table rows to re-order the circuit pins.
6. (Optional) The results of the post processing calculation (defined on the Matrix dialog Post Processing tab) on the matrix can be included by selecting **PostProcessed.**
7. In the **Circuit Export** section, select a setup from the **Solution Setup** pull-down list.

8. (Optional) Click the **...** button to open the **Set Design Variation** dialog box to choose a **Variation**.
9. Type the path and file name in the **File name** text box, or click the **...** button to select the desired location for the exported circuit.
10. (Optional) To view the netlist:
    a. Click **Preview**.
       The **Circuit Model Preview** dialog box appears. You can also click **Export Circuit** from within this dialog box.
    b. Click **Close** to close the netlist.
11. To export the circuit to the specified file, click **Export Circuit**.
12. Click **OK** to close the **Export Circuit** dialog box.
13. Click **Close** to close the **Solutions** dialog box.

Coupling Coefficient Matrix

For a matrix with entries M_{ij}, i, j = 1,..., N, the coupling coefficient for row i and column j is given by the following equation:

$$k = \frac{M_{ij}}{\sqrt{M_{ii} \times M_{jj}}}$$

This coefficient indicates how much flux in one coil is linked with the other coil.

If all the flux in one coil reaches the other coil, $|k| = 1$ (and coupling is 100%).

Inductors with $|k| > 0.5$ are tightly coupled. Inductors with $|k| < 0.5$ are loosely coupled.

The mutual inductance (*M*) falls between 0 and $\sqrt{M_{ii} \times M_{jj}}$ (the geometric mean of the inductances).

Related Topics

Assigning a Matrix

AC Loss Effects of Litz Wire on Impedance Matrix

The impedance matrix of poly-phase windings is expressed as:

$$[Z] = \begin{pmatrix} R_{11} + jX_{11} & \cdots & R_{1n} + jX_{1n} \\ \vdots & \ddots & \vdots \\ R_{1n} + jX_{1n} & \cdots & R_{nn} + jX_{nn} \end{pmatrix}$$

where *n* is the number of windings.

Litz wire eddy current losses are computed in a post-process. When the total number of windings includes some Litz wire windings, the final AC impedance matrix, including an additional resistance matrix due to Litz wire eddy current losses, is expressed as:

$$[Z'] = \begin{pmatrix} R_{11} + jX_{11} & \cdots & R_{1n} + jX_{1n} \\ \vdots & \ddots & \vdots \\ R_{1n} + jX_{1n} & \cdots & R_{nn} + jX_{nn} \end{pmatrix} + \begin{pmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{1n} & \cdots & r_{nn} \end{pmatrix}$$

where resistances denoted by r_{ij} are additional Litz wire eddy current resistance components.

To compute Litz wire eddy current resistance components, we use a permeability freezing algorithm. After nonlinear iteration is convergent and permeabilities in all mesh components are frozen, the original impedance matrix *[Z]* is derived by injecting 1A exciting current into each phase winding one by one. The self-impedance is obtained from the induced voltage in the excited winding, and all mutual impedances are obtained from the induced voltages in all other windings.

However, the Litz wire eddy current resistance matrix is derived in a different way. When a winding is excited with 1A rms current and all other windings with zero-value currents, the value of the total eddy current loss (no matter in the excited winding, or in other zero current windings), is the value of the self-resistance. That is, if I_1 = 1A rms current in winding 1 produces total eddy current loss P_1, then the self-resistance of winding 1 is:

$$r_{11}{I_1}^2 = r_{11} = P_1$$

Similarly, if I_2 = 1A rms current in winding 2 produces total eddy-current loss P_2, then the self-resistance of winding 2 is:

$$r_{22}{I_2}^2 = r_{22} = P_2$$

To calculate the mutual resistance between winding 1 and winding 2, we need to inject 1A currents in both winding 1 and winding 2. In such a case, the total eddy-current loss is:

$$r_{11}{I_1}^2 + 2r_{12}I_1I_2 + r_2{I_2}^2 = r_{11} + 2r_{12} + r_{22} = P_{12}$$

where P_{12} is the total eddy current loss in all Litz wire windings caused by 1A currents in both winding 1 and winding 2. Therefore,

$$r_{12} = \frac{P_{12} - r_{11} - r_{12}}{2}$$

Related Topics

Assigning a Matrix

Viewing Parameter Data

To view parameter solution data such as force and torque:

1. In the project tree, right-click the solution of interest in the **Parameters** folder, and then click **View Solution** on the shortcut menu.

   The **Solutions** dialog box appears.

2. The **Design Variation** box shows the current design.

   Optionally, click the **[...]** button, and choose a design variation solved during an optimization or parametric analysis from the **Set Design Variation** dialog box.
3. In the **Simulation** pull-down lists:
   - Select the solution setup from the first pull-down list.
   - Select the solved pass data you wish to view from the second pull-down list, either the last adaptive pass or one of the intermediate passes.
4. On the desired parameter tab, such as **Force** or **Torque**, select a **Parameter** and optionally an adaptive **Pass** to view.

   Depending on the solution type you selected, you may have to specify the **Units** in which to display the information. The available units depend on the matrix type being displayed.

   Optionally, to export the parameter solution data:
   a. Click **Export Solution** to open the **Export Solution** dialog box appears.
   b. Select a location for the solution data.
   c. Type a filename in which to store the data in the **File name** text box.
   d. Click **Save**.
5. Click **Close** to close the **Solutions** dialog box.

Related Topics

Adding solution Parameters

Viewing Mesh Statistics

To view an adaptive solution's mesh information, either during or after the solution process:

1. In the project tree, right-click the solution setup of interest, and then click **Mesh Statistics** on the shortcut menu.

   The **Solutions** dialog box appears with the **Mesh Statistics** tab selected.

   The table lists the design elements and depending upon the solution type may include: Num Tets (tetrahedra), Min edge length, Max edge length, RMS edge length, min tet vol., max tet vol., mean tet vol. and standard deviation (min elem area, max elem area, mean elem area and standard deviation for 2D designs).

   If mesh repairs have been performed, two additional columns appear in the table; Recovered % and Reparied %. These columns indicate the fraction of an object that was successfully recovered and the fraction that needed some repair.

To toggle the mesh statistics display from low to high values or visa versa:

1. Click on the column header.

   This displays a shadowed triangle pointing down to indicate a list ordered from highest to lowest, and a triangle pointing up to indicate a list ordered from lowest to highest. Clicking again inverts the current order.

Click on the blank cell above the object list to invert the order of objects, though in this case, the cell does not display a directional triangle.

Related Topics

Technical Notes: Finite Element Analysis

Technical Notes:The Mesh Generation Process

Viewing Winding Results (Eddy Current)

To view winding results for Eddy current solutions:

1. In the project tree, right-click the **Results** folder, and then click **Solution Data** on the shortcut menu.

   The **Solutions** dialog box appears.
2. The **Design Variation** box shows the current design.

   Optionally, click the **[...]** button, and choose a design variation solved during an optimization or parametric analysis from the **Set Design Variation** dialog box.
3. In the **Simulation** pull-down lists:
   - Select the solution setup from the left pull-down list.
   - Select the solved pass data you wish to view from the right drop-down list – either the **LastAdaptive** pass, or an intermediate **AdaptivePass**. If LastAdaptive is selected, the **Freq** drop-down list is enabled for you to choose frequency. If AdaptivePass is selected, the **Pass** drop-down list is enabled for you to choose the desired intermediate pass.
4. If the design includes windings, the **Winding** tab displays the following columns: **Flux Linkage[Wb]**, **Induced Voltage[V]**, **Input Current[A]**, **Input Voltage[V]**, and **Current[A]**. Results listed in these columns are complex quantities. The **Format** tab allows you to customize how values are displayed by setting the desired width, precision, and whether to use scientific notation.
   - For Current type windings: Flux Linkage, Induced Voltage, and Input Current results are shown. The other quantities are shown as N/A (not applicable).
   - For Voltage type windings: Flux Linkage, Induced Voltage, Input Current, and Current results are shown. The other quantities are shown as N/A (not applicable).
   - For External type windings: Flux Linkage, Induced Voltage, and Current results are shown. The other quantities are shown as N/A (not applicable).
5. Click **Close** to close the **Solutions** dialog box.

Viewing End Connection Results (2D Eddy Current)

To view end connection results for Maxwell 2D Eddy current solutions:

1. In the project tree, right-click the **Results** folder, and then click **Solution Data** on the shortcut menu.

The **Solutions** dialog box appears.

2. The **Design Variation** box shows the current design.

   Optionally, click the **[...]** button, and choose a design variation solved during an optimization or parametric analysis from the **Set Design Variation** dialog box.
3. In the **Simulation** pull-down lists:
   - Select the solution setup from the left pull-down list.
   - Select the solved pass data you wish to view from the right drop-down list – either the **LastAdaptive** pass, or an intermediate **AdaptivePass**. If LastAdaptive is selected, the **Freq** drop-down list is enabled for you to choose frequency. If AdaptivePass is selected, the **Pass** drop-down list is enabled for you to choose the desired intermediate pass.
4. If the design includes end connections, the **End Connection** tab displays columns for: **Induced Voltage[V]**, and **Current[A]**. Results listed in these columns are complex quantities. The **Format** tab allows you to customize how values are displayed by setting the desired width, precision, and whether to use scientific notation.
5. Click **Close** to close the **Solutions** dialog box.

Viewing Loss Results (Eddy Current)

To view loss results for Eddy current solutions:

1. In the project tree, right-click the **Results** folder, and then click **Solution Data** on the shortcut menu.

   The **Solutions** dialog box appears.
2. The **Design Variation** box shows the current design.

   Optionally, click the **[...]** button, and choose a design variation solved during an optimization or parametric analysis from the **Set Design Variation** dialog box.
3. In the **Simulation** pull-down lists:
   - Select the solution setup from the left pull-down list.
   - Select the solved pass data you wish to view from the right drop-down list – either the **LastAdaptive** pass, or an intermediate **AdaptivePass**. If LastAdaptive is selected, the **Freq** drop-down list is enabled for you to choose frequency. If **AdaptivePass** is selected, the **Pass** drop-down list is enabled for you to choose the desired intermediate pass.
4. The **Loss** tab displays the following columns: **Core Loss[W]**, **Solid Loss[W]**, **Stranded Loss[W]**, **Stranded Loss R[W]**and **Stranded Loss AC[W]**. Results listed in these columns are real quantities. The **Format** tab allows you to customize how values are displayed by setting the desired width, precision, and whether to use scientific notation.
5. Click **Close** to close the **Solutions** dialog box.

Cleaning Up Solutions

You can use **Clean Up Solutions** to selectively make deletions, or remove all solutions from the results.

To use **Clean Up Solutions**:

1. Depending upon the project type, click **Maxwell3D, Maxwell2D, or RMxprt and then select Results>Clean Up Solutions**.

   The **Clean Up Solutions** dialog box appears.
2. Under **Solutions**, select whether you want to delete only fields data, only fields and mesh data, only linked data, or all solution data. Deleting all solution data erases all mesh, matrix, and fields data for all adaptive passes and frequency sweeps for the selected **Variations**.

   Optionally, you can **Include Linked Data** in the deletions.

   Linked data can be mesh, field or some other post-processing data that the source design generated. The target design for the link caches these data internally to minimize the need to activate the source design.
3. Under **Variations**, select which solution data you want to delete:
   - Select **All Except Current Variation** to delete all solution data that do not correspond to the current project and design variable values for the current design.
   - Select **All Variations** to delete all solution data for the current design.
   - Select **Select** to specify the variations you wish to delete. Click **Variations** to select the variations for deletion.
4. Click **Do Deletions**.

   The solution data you selected are deleted. Any post processing reports or field overlays you created that included data you deleted will be marked with an X in the project tree. They will be invalid until new solution data are generated.

Related Topics

Monitoring the Solution Process

Deleting Reports

Creating Animations

An animated plot is a series of frames that displays a field, mesh, or geometry at varying variable values. To create an animated plot, specify the variable values of the plot that you want to include. Each plot associated with a variable value is a frame in the animation. You specify how many frames to include in the animation.

Note	Each animation frame requires memory for storage which depends upon the mesh size and type of plot. Memory usage may become very large during plot animations. To reduce memory usage, specify the minimum number of frames possible. See General Options for more information.

You can export the animation to animated Graphics Interchange Format (GIF), to Audio Video Interleave (AVI) format, or to WebM format.

Related Topics

General Options: Desktop Performance
Creating Geometry Animations
Creating Field Plot Animations
Controlling the Animations Display
Exporting Animations

Creating Geometry Animations

Geometry animations may be created to evaluate the effect of varying geometry variables on the model. You must define at least one variable associated with the geometry prior to creating a geometry animation. Following is the general procedure for creating an animation that varies a part of the model geometry:

1. Right-click in the view window, then click **View>Animate**.
   If multiple geometries can be varied in the design, the **Select Drawing** dialog box appears, proceed to step 2. If only one geometry is variable, proceed to step 3.

   The **Setup Animation** dialog box appears.
2. In the **Select Drawing** dialog box:
   a. Select the geometry variable to vary in the animation.
   b. Select the object you want to animate.

Note	If previous animations have been created for this project, the **Select Animation** dialog will appear. You may choose an animation setup from the list if one is associated with the geometry variable of interest and the animation will start. If no existing animation setup is acceptable, select **New** and continue at Step 3 below.

3. In the **Setup Animation** dialog box:
   a. Type a name for the animation in the **Name** text box or accept the default name.
   b. Optionally, type a description of the animation in the **Description** text box.
   c. Under the **Swept Variable** tab, the **Swept Variable** list includes all of the defined geometric project and design variables. Select the geometry variable that you want to animate from the **Swept Variable** list.
   d. Specify the values of the variable that you want to include in the animation:
4. Type the starting value of the variable in the **Start** text box.
5. Type the stopping value of the variable in the **Stop** text box.
6. Type the number of **Steps** to include in the animation.
   For example, if the **Start** value is **0.15in**, the **Stop** value is **0.45in**, and the number of steps is **15**, the animation will display the geometry at 15 values between 0.15 inches and 0.45 inches. The animation will also include the start value, which will be the first frame displayed, resulting in a total of 16 frames in the animation.

a. If the design has multiple project or intrinsic variables, click the **Design Point** tab to set the values of the non-animated variables.

4. Click the **Design Point** tab.
5. Deselect the **Use defaults** check box.
6. In the table, select the row corresponding to the variable setting of interest.
7. Click **OK**.

   The animation begins in the view window, displaying one frame for each variable value.

   The play panel appears in the upper-left corner of the desktop, enabling you to stop, restart, and control the speed and sequence of the frames.

Related Topics

Controlling the Animation's Display

Creating Field Plot Animations

To animate an existing field plot:

1. Right-click on the plot name in the **Project Tree**, and then select **Animate**.
   The **Setup Animation** dialog box appears.

Note	If previous animations have been created for this project, the **Select Animation** dialog will appear. You may choose an animation setup from the list if one is associated with the geometry variable of interest and the animation will start. Otherwise select **New** and continue at Step 2 below.

2. Type a name for the animation in the **Name** box, or accept the default name.
3. Optionally, type a description of the animation in the **Description** box.
4. Under the **Swept Variable** tab, the **Swept Variable** list includes all of the defined project and design variables. Select the variable that you want to animate from the **Swept Variable** pull-down list.
5. The **Select Values** table lists all solved values for the variable selected in **Swept Variable**. Select the values to include in the animation. By default, all solved values will be included.
6. *(For Electric Transient Solver plots only)*
   a. Type the starting value of the variable in the **Start** box.
   b. Type the stopping value of the variable in the **Stop** box.
   c. Type the number of **steps** to include in the animation in the **Steps** box.
      For example, if the **Start** value is **0.15in**, the **Stop** value is **0.45in**, and the step size is **15**, the animation displays the geometry at 15 values between 0.15 inches and 0.45 inches. The animation also includes the start value, which is the first frame displayed, resulting in a total of 16 frames in the animation.
7. If the design has multiple project or intrinsic variables, click the **Design Point** tab to set the values of the non-animated variables.

a. Deselect the **Use defaults** check box.
b. In the table, select the row corresponding to the variable setting of interest.
c. If intrinsic variables are available, select the value of interest in the **Intrinsic variables** section of the dialog.

8. Click **OK**.

   The animation begins in the view window, displaying one frame for each variable value.

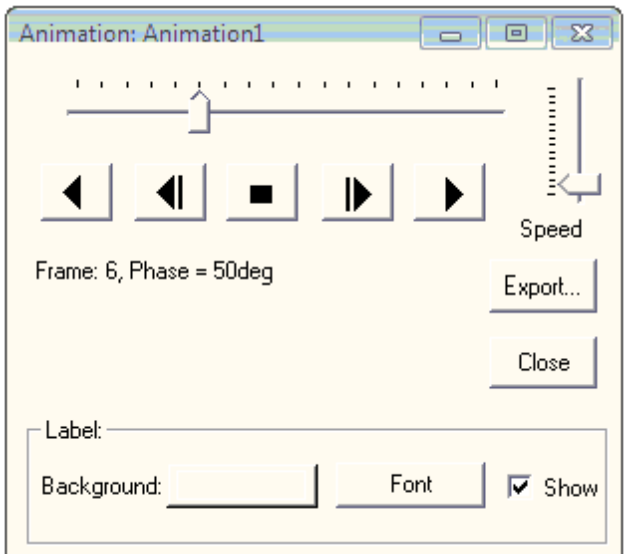
   The play panel appears in the upper-left corner of the desktop, enabling you to stop, restart, and control the speed and sequence of the frames.

Related Topics

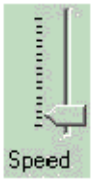
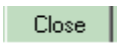
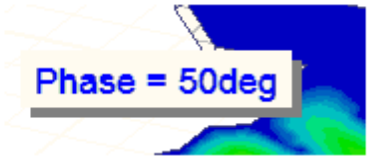
Controlling the Animation's Display

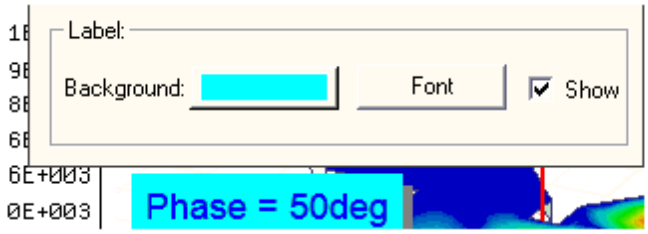
Controlling the Animation's Display

When an animation is displayed in the view window, the **Animation** window, also called the *play panel*, appears in the upper-left corner of the desktop. It has buttons that enable you to control the speed and sequence of the frames, start and stop the animation and export the animation. Click an area of the window below to learn its function.



Animation slider	Each dot on the slider represents a frame in the animation. Drag the slider to the right to display the next frame in the animated plot. Drag the slider to the left to display the previous frame in the animation.
	Plays the plot's animation sequence backwards.

	Steps backward through the animated plot one frame at a time.
	Stops the animation.
	Steps forward through the animated plot one frame at a time.
	Plays the plot's animation sequence forwards.
	Drag the **Speed** slider to the top to increase the speed of the animation. Drag the **Speed** slider to the bottom to decrease its speed.
Frame information	The current frame and phase at which the plot is being displayed is listed below the control buttons.
	Enables you to export the animation to an animated Graphics Interchange Format (GIF) or to Audio Video Interleave (AVI) format.
	Closes the animation window.
Show label checkbox	If you select the Show checkbox, a label showing the swept variable value appears in the animation. You can select the label with the mouse and drag it to another location. 
Background	Click the background button to open a color palette dialog that lets you set the

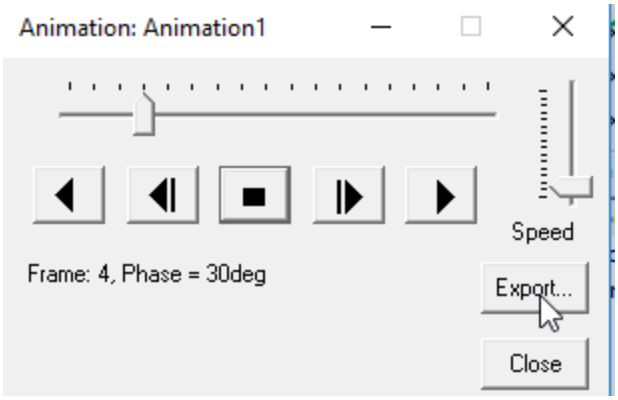
	background for the swept variable label. 
	The font button opens a font selection dialog that you can use to set the Font, Font Style, and Size for the label. The Default is Arial Narrow 14pt.

Related Topics

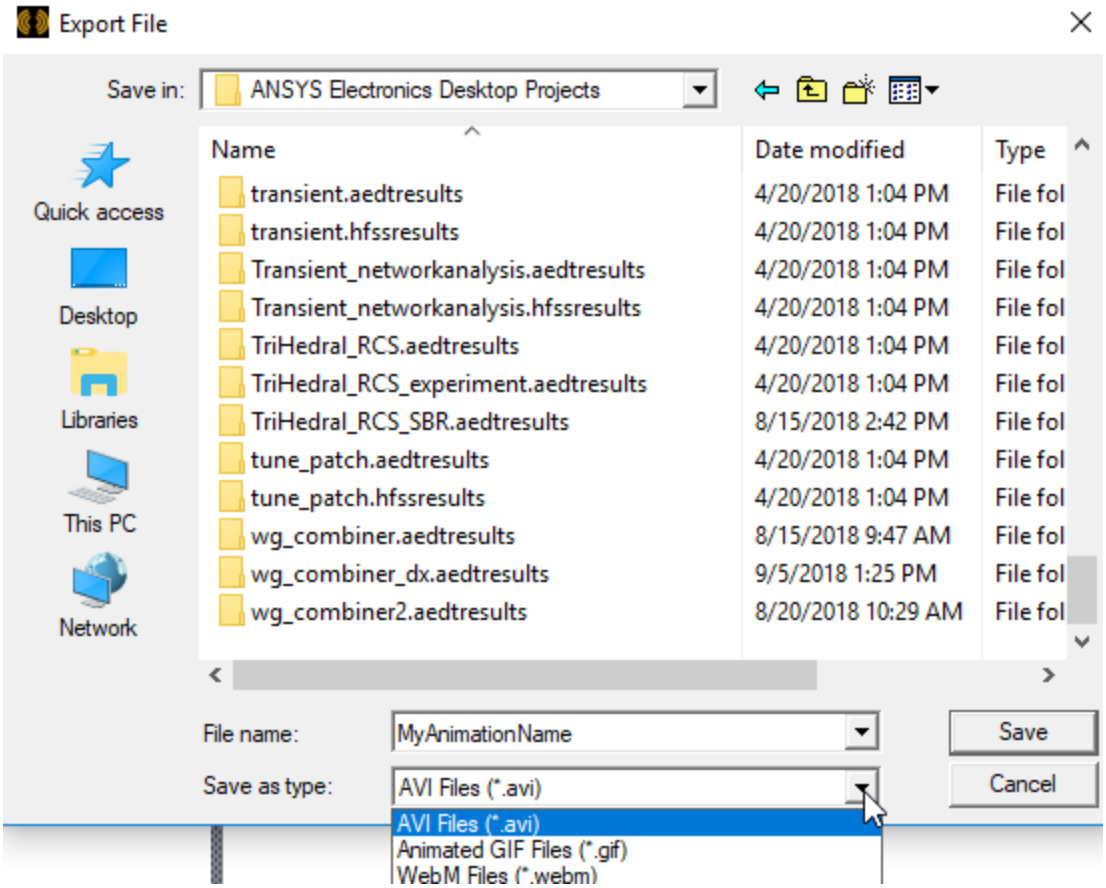
Creating Animations

Exporting Animations

1. Create the animation you want to export.
2. In the play panel, click **Export**.

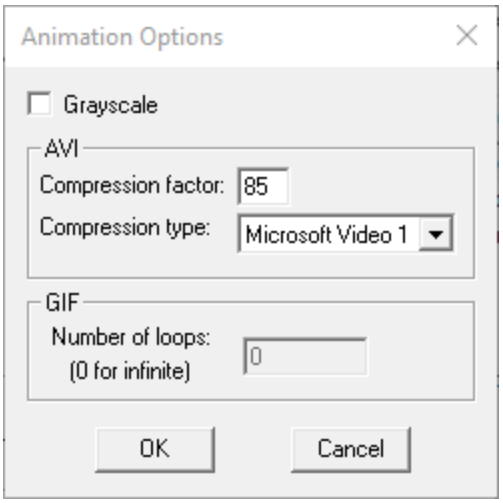


The **Export File** dialog box appears.



3. Specify the directory to **Save in**, the **File name** and use the **Save as type** drop down men to select **Animated GIF File (.gif)** or **AVI File (.avi)** or **WebM File (.webm)**.

   The **Animation Options** dialog box appears.



4. To replace colors in the file with 256 shades of gray, select **Grayscale**.

Grayscale animations tend to use less memory than full color animations.

5. For AVI format export, specify the **Compression factor** (the default is 85) and one of the following **Compression types**:

INTEL Indeo	
Cinepak	
Microsoft Video 1	
None	

6. For GIF format export, specify the number of loops. The default "0" denotes infinite loops.
7. Click **OK** to close the **Animations Options** dialog.

   The animation is exported to the file format you specified.

Creating Reports

After Maxwell has generated a solution, all of the results for that solution are available for analysis. One of the ways you can analyze your solution data is to create a 2D or 3D report, or graphical representation, that displays the relationship between a design's values and the corresponding analysis results. You create reports using either the **Create Quick Report** command, or the **Create<*type*> Report** commands. The **Quick Report** feature lets you select from a list of predefined categories (such as S-parameters) from which to create a rectangular plot.

For each solution type – Electrostatic, Magnetostatic, AC Conduction(2D only), DC Conduction, Eddy Current, Transient, Transient A-Phi Formulation (3D only), or Electric Transient(3D only) – the **Results** menus present a list of **Create <*type*> Report** commands based on the solution data of direct interest for the design. For example, for the Magnetostatic solution type, the **Results** menu contains templates for Magnetostatic parameters and for Fields.These appear on the menus as **Create Magnetostatic Report** and **Create Fields Report**. Each of these **Create <*type*> Report** menu items includes a further cascading menu that lists the Display Types available for that report. For some reports you can modify the Display Type from the Properties for that Report.

If you have created custom report templates (for example, including your company name or other format changes), you can also create a report based on that template by selecting **Maxwell 2D, Maxwell 3D, or RMxprt>Results>Report Templates>**<*templateName*>. You can also access previously defined templates using **Report2D>Report Templates>Apply Settings**. You can save the properties for a modified report to provide the custom default settings for all new reports.

You can also use the **Report2D>Export** feature and select ReportData File (.rdat) format file which you can then select for **Create Report from File**.

Related Topics

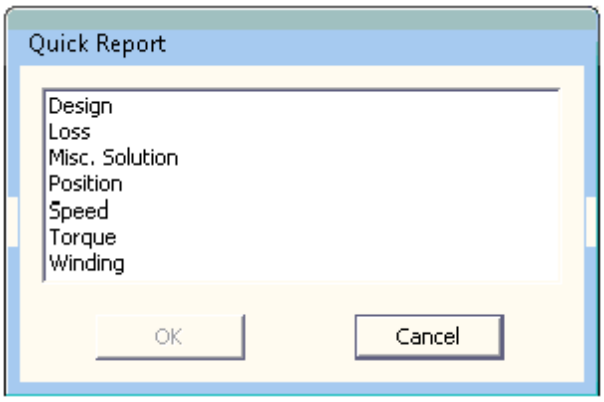
Creating a Quick Report

Creating a New Report

Creating a Quick Report

Following is the procedure for creating a quick report.

1. On the Project tree under "Analysis", select a **setup** or **sweep icon**, or the **Results** icon.
2. Right-click to display the shortcut menu and select **Create Quick Report**.
   The **Quick Report** dialog appears.



3. Select the one or more categories for the report from the list and click OK. The list of categories varies with the solution type.

A rectangular plot for each selected category displays. The new plot or plots appear in the Project tree under the Results icon. The default Report Name that appears is derived from the report category specified in the **Quick Report** dialog box. You can edit the plot names in the project tree and the plot header text in the report synchronizes.

Related Topics

Creating Reports
Modifying Reports
Quick Report Categories
Creating Quick Reports in RMxprt

Quick Report Categories

When using the Quick Reports function for Solutions, the following report categories may be available depending upon the solution parameters requested, solution type, etc:

Category	Description
Coupling Coeff	Reports values of the coupling coefficient.
Design	Reports values of design variables.
Force	Reports values of Force parameters such as Force Magnitude and Components, Load Force.
Torque	Reports values of Torque parameters such as Torque Magnitude and Components
L, Lnom	Reports values of Inductance Matrix calculations.
MagFlux, MagFluxNom	Reports values of magnetic flux calculations.
C	Reports values of Capacitance Matrix calculations.
R	Reports values of Resistance Matrix calculations.
Z	Reports values of Impedance Matrix calculations.
Loss	Reports values of Loss parameters such as CoreLoss, EddyCurrentLoss, ExcessLoss, HysteresisLoss, SolidLoss, StrandedLoss, StrandedLossR, and StrandedLossAC.
Misc Solution	Reports values for the solution process such as DeltaT, Energy Error, and Temporal Error
Position	Reports Position of moving objects.
Speed	Reports Speed of moving objects.
Winding	Reports Flux Linkages, currents and voltages in coils.

Creating a New Report

Following is the general procedure for creating a new report:

1. On the **Maxwell3D** or **Maxwell2D** menu or the Project tree, point to **Results**, and then select **Create <*type*> Report** and select **Display Type** for that template. The available **Report Types** depend on the simulation setup.

   If you have created custom report templates (for example, including your company name or other format changes), you can also create a report based on that template by selecting **Maxwell 2D, Maxwell 3D, or RMxprt>Results>Report Templates>PersonalLib>**<*templateName*>. You can also make such changes the default for new reports by right-clicking a modified report and selecting **Report Templates>Save Settings as default**.

   When you have selected the ***<type>*** and display type from the **Results** menu, the **Report** dialog box appears, with the **Trace** tab selected by default.

2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   - **Solution** field with a drop down selection list. This lists the available setups and sweeps. As a minimum, the LastAdaptive solution and AdaptivePass solution is available to choose.

     The AdaptivePass solution context can be selected to allow any value or parameter to be plotted versus the adaptive pass. This function is usually used to evaluate the convergence of the solution.
   - *For Transient projects only*:

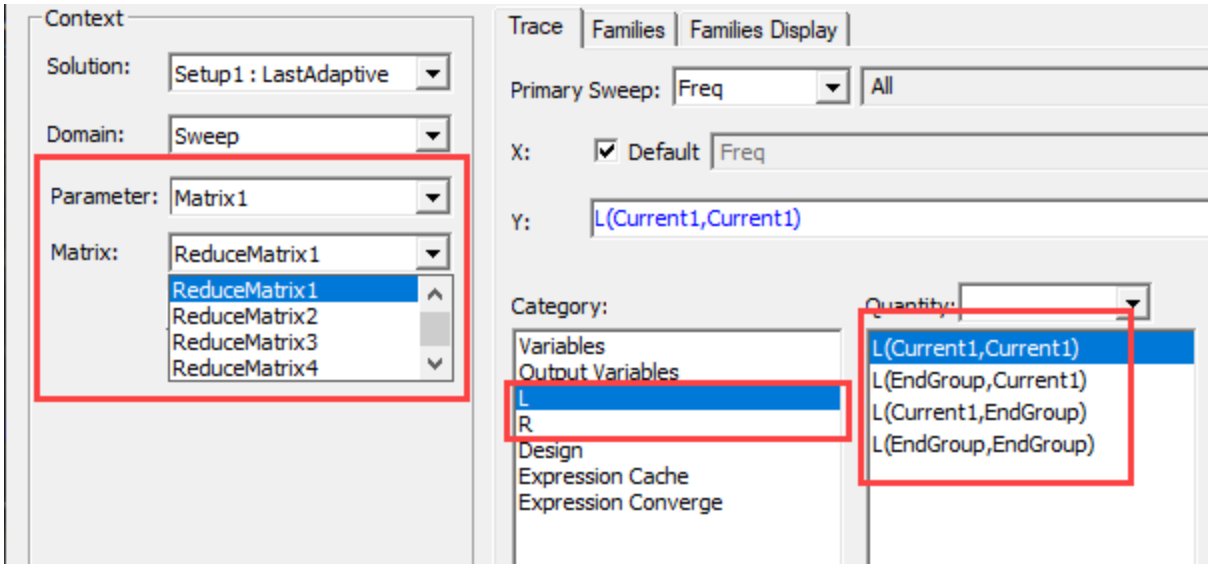
     Domain field with a pull down selection list containing options for plotting vs time (Sweep Domain) or plotting vs frequency (Spectral Domain).
   - *For Transient projects only*:

     Domain field with a pull down selection list containing options for **Average and RMS** and **Transient D-Q** for electric machines. Click **Machine Options** to setup the machine specific parameters.
   - *For Eddy Current projects only*:

     Domain field with a pull down selection list containing options for plotting vs frequency (Sweep Domain) or plotting vs time (Time Domain).

     Matrix field with a pull down selection list containing options for plotting *matrix* and *reduce matrix* parameters.

Note	In Maxwell Eddy Current designs, the user can create *matrix* parameters, which will cause the solver to produce an impedance *matrix* for the selected excitations. In addition, the user can group (wire) two or more excitations to one excitation in either a series or parallel connection referred to as a *reduce matrix*. The **Matrix** field appears only if a matrix entry is selected in the **Parameter** field. (Refer to *Assigning a Matrix* for information on creating a *matrix* and *Assigning a Reduce Matrix* for information on creating *reduced* matrix parameters for Eddy Current designs.)



- **Parameter** field with a drop down selection list. Whether this field appears, and the parameters listed depend on the Solution type and the ***<type>*** selected.
- *For Fields and Noise Vibration reports*: **Geometry** field with a drop down selection list. This applies the quantity to a specific geometry.

3. In the **Y** Component section, make selections for the following:
   a. Categories - depend on the Solution type and the design. For example, Magnetostatic categories include: Torque, Output Variables, Inductance, and other user-selectable solution parameters. Transient categories include: Loss, Output Variables, Variables, and others. Eddy current categories include: Loss, Winding, End Connection, L (inductance), R (resistance) Z (impedance), and others.
   b. Quantities for Y are relative to the selected category. For example, Loss quantities include: CoreLoss, EddyCurrentLoss, HysteresisLoss, and others. Winding quantities include: Flux Linkage, Induced Voltage, and others - depending on the winding type.

Note	The Quantity text field can be used to filter the Quantity list by typing in text, or by using the four predefined selections. This is useful if the Category selected produces a lengthy Quantities list. See Filtering Quantity Selections for the Reporter. When the matrix is very large, the number of quantities can be correspondingly huge. Therefore, the Quantities field can optionally use a tree structure to divide matrix quantities into groups by their first element name. The initial display shows groups, without initially listing group members.

   c. **Function** list to apply to the Y quantities.
   d. The Y value field displays the currently specified Quantity and Function. You can edit this field directly.

Note	Color shows valid expression.

e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.

4. In the **X (Primary Sweep)** section, make selections for the following:
   a. Select the Primary sweep from the drop down menu. By default **All** of the chosen sweep's values are used. You can also select the browse [...] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time sweeps), or **Use all values** (the default setting).
      To select an X component that is different from the Primary Sweep, uncheck the Default field to enable the X field and browse [...] button. Click the browse [...] button to display the **Select X Component** dialog box. This lets you specify the X component as you do the Y; that is, in terms of Categories which define the selectable Quantities, and Functions to apply. After making selections, **OK** the dialog to assign the X component.
   b. The **Families** tab provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined (for example, for a parametric sweep). If so, the variables other than the one chosen as the **X (Primary sweep)**, are listed under the **Families** tab with columns for the variable, the value, and an Edit column with an ellipsis [**...**] button. See Using Families tab for Reports.
5. **Update Report** setting
   - **Real Time** checked -- enable real time updates for all reports while the reports are being edited.
   - **Real Time** unchecked -- enables drop down menu to **Update All Reports** or **Update Report**. Reports will only be updated with one of these user selectable update options or upon exiting the report dialog box. This can be useful if you expect a trace to take time to display. You can then add additional traces without having to wait.
6. The **Report** dialog command buttons permit you create a new report with the settings you provide, or to modify an existing report.
   - **Output Variables** – opens the **Output Variables** dialog box.
   - **Add Trace** – this is enabled when you have created or selected a report. Add one or more traces to include in the report.
   - **Update Trace** – updates the selected traces in a report based on further processing or changes.
   - **New Report**. Adds a report to the Project tree under the Results icon. The new Report is displayed in the Project window.
   - **Options** – opens the **Report Setup Options** dialog box. This contains a check box for using the advanced mode for editing and viewing trace components. This mode is automatic if the trace requires it. It also contains a field for setting the maximum number of significant digits to display for numerical quantities.
   - **Close** – closes the **Report** dialog box.
7. Click **New Report** to create a new report in the Project tree.

   The report appears in the view window. It will be listed in the project tree under Results, with the default name based on the Report Category you selected, for example, Force Plot *n* or Output Variables Plot *n*. You can edit the plot names in the project tree and the plot header text in the report synchronizes. Traces within the report also appear in the project tree. Some plots may take time to complete. Performing a **File>Save** in such cases after the plot

has been created will permit you to review the plot later without having to repeat the calculation time when you reopen the project later.

8. To speed redraw times for changed plots, perform a **Save**. This saves the data that comprises expressions. If you do not do a save of a changed plot, the changed version is not stored.

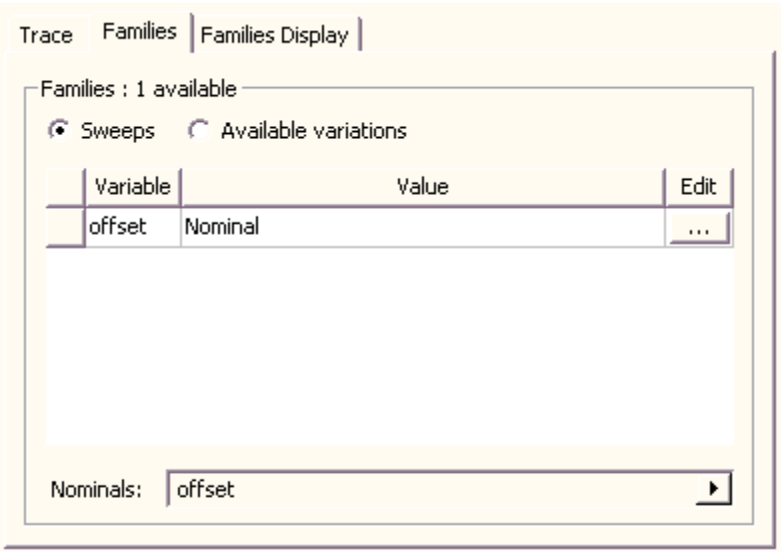
Note	Remember the evaluated value of an expression is always interpreted as in SI units. However, when a quantity is plotted in a report, you have the option to plot values in units other than SI. For example, the expression "1+ang_deg(S_{11})" represents an 'angle'' quantity evaluated in radians, though plotted in degree units. To represent an angle quantity in degrees, you would specify units as "1 deg + ang_deg(S_{11})".

Related Topics

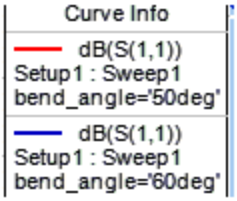
Creating Reports
Modifying Reports
Creating a Quick Report
Using Families tab for Reports
Plotting a Quantity versus Adaptive Pass

Using Families Tab for Reports

The **Families** tab of the Report dialog provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined and solutions exist for multiple variable values (for example, for a parametric sweep). If no variables are defined, or none have solutions for different values, 0 families will be available. If so, the variables other than the one chosen as the **X (Primary sweep)**, are listed under the **Families** tab with columns for the variable, the solution value (which may be All, Nominal, or a Specific value), and an Edit column with an ellipsis [**...**] button. Families gives the number available. If an existing variable is specified as Nominal, only that value is currently available. You can set any solved variables as Nominal, All, or select from values provided for Available solutions.



When you select a variable with multiple solved values, a trace for each solved value appears in the Report, with the variable value appended to the trace name in the Report legend.



You can make selections for the following.:

1. Select the **Sweeps** radio button (the default) to list the swept variables you can select.
2. With the **Sweeps** radio button selected, click the ellipsis [**...**] button to display a list of variable values for a particular variable. You can use a scroll bar to navigate the list.
   - To select all values, click the check box for **Use all** values. This writes “All” in the value field for that variable. You can also select individual values by clicking on them.
   - To select a range of values, hold down the shift key, and click again.
   - To select intermittent additional values, hold the CTRL key and click additional. The values you select are highlighted in the list, and are also listed in the Values column for that variable.
   - To select all, use the **Select All** button. This highlights the complete list, as well as listing all values for the variable in the Value field.
   - To clear the selections, use the **Clear All** button.
3. To edit the available list, click the ellipsis [**...**] button to display the **Edit Sweep** dialog. For the selected variable, this lets you specify:

- a single value as value and units
- a linear step with start value, stop value, and step value, and units for each.
- a linear count, with start value, stop value with units, and a count.
- a decade count
- an octave count
- an exponential count.

Once you have specified the changes you want, you can add you change to the list by clicking **Add>>** or replace the current list by clicking **Update>>**.

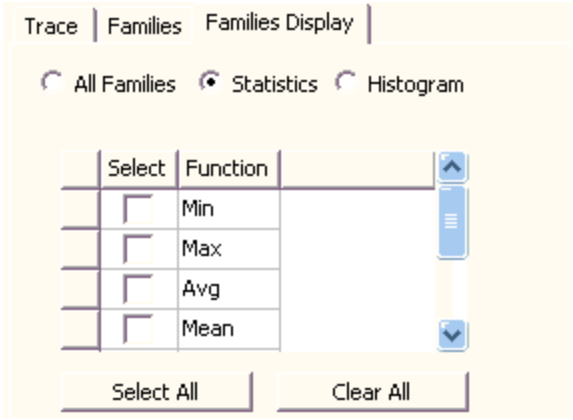
4. Click **OK** to close the **Edit Sweep** dialog box. You must still select the **Edited** radio button to be able to select the from the edited list, rather than the Default selection. And you must make selections here to move the new values to the variable value list in the **Families** tab.

   .Select the **Available variations** radio button to list the choices that derive from variable combination.

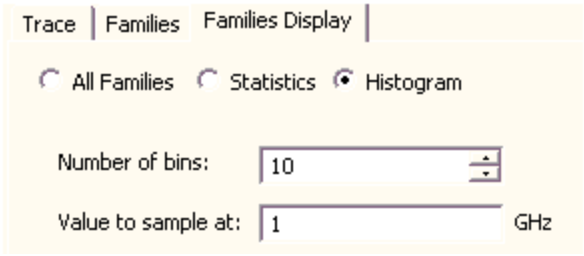
   a. To select individual variations, check the select box.
   b. Click the **Select** button at the top of the column to check or clear all variations at once.
   c. To invert the list order, click the triangle beside the variable name.

   The **Families Displa**y tab has three radio button selections.

   a. **All Families**
   b. **Statistics** which lists a table statistical functions that you can select to apply to the plot. The functions include Min, Max, Avg, Mean, Variance, Std Dev, and Sum. You can use the Select check boxes or the Select All and Clear All button.



   c. **Histogram** which lets you select the number of bins to use for a histogram plot, and the sampling frequency to use.

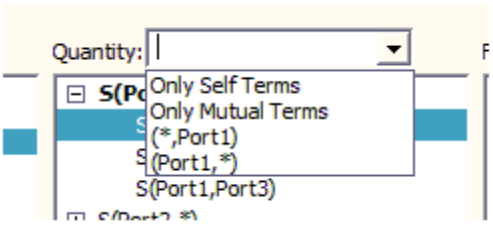


Related Topics

Creating Reports
Modifying Reports
Modifying the Background Properties of a Report
Modifying the Legend in a Report
Creating Custom Report Templates
Working with Traces
Editing the Display Properties of Traces
Discarding Report Values Below a Specified Threshold
Adding Trace Characteristics
Adding Data Markers to Traces

Filtering Quantity Selections for the Reporter

When a two port quantity Category is selected, four predefined filters are added to the combo box. “Port1” is the first matrix element name found in the quantity list.



- Only Self Terms -- Only display quantities when the first and second port are same.
- Only Mutual Terms -- Only display quantities when the first and second port are different.
- (*,Port1) -- Only display quantities when the second element name is “Port1”. You can edit the element name to display quantities for other elements.
- (Port1,*) – Only display quantities when the first element name is “Port1”. You can edit the element name to display quantities for other element.

Creating a Report from a Report Data File

Using the Report Data File format, **.rdat**, a report can be saved (exported) from one design, and subsequently imported into a different design (or even imported into the same design). This allows you to take a static snapshot of a set of simulation data and view it at a later stage of the design process, or view the same data set in one or more different designs.

To save a report to a Report Data File:

1. Create the report that you wish to save.
2. Right-click in the report window (or right-click on the desired report icon in the project tree) and select **Export** to open the **Export Report** dialog.
3. Select the Report Data files option (**.rdat**) in the **Save as type** selection drop down.
4. Navigate to the desired location and either create a file name or select an existing file for the exported report; and then click **Save**.

To create a report from an existing Report Data File:

1. Right-click on the **Results** folder in the project tree and select **Create Report From File**.
2. Browse to select the desired Report Data file (**.rdat**), and then click **Open**.
3. The selected report is added to the design and opens in a new report window.

Modifying Reports

To modify the data that is plotted in a report:

1. In the project tree, click the report you want to modify.
2. Right-click **Modify Report**.

   The **Report** dialog appears.
3. The **Report** dialog command buttons permit you create a new report with the settings you provide, or to modify an existing report.
   - **Output Variables** – opens the **Output Variables** dialog.
   - **Add Trace** – this is enabled when you have created or selected a report. Add one or more traces to include in the report.
   - **Update Trace** – updates the selected traces in a report based on further processing or changes.
   - **New Report**. Adds a report to the Project tree under the Results icon. The new Report is displayed in the main window.
   - **Options** – opens the **Report Setup Options** dialog box. This contains a check box for using the advanced mode for editing and viewing trace components. This mode is automatic if the trace requires it. It also contains a field for setting the maximum number of significant digits to display for numerical quantities.
   - **Close** – closes the **Report** dialog.

   The updated report appears in the view window.
4. **Update Report** setting

- **Real Time** checked -- enable real time updates for all reports while the reports are being edited.
- **Real Time** unchecked -- enables drop down menu to **Update All Reports** or **Update Report**. Reports will only be updated with one of these user selectable update options or upon exiting the report dialog box. This can be useful if you expect a trace to take time to display. You can then add additional traces without having to wait.

5. In the **Context** section you make selections depending on the design and solution type.
   - **Solution** field with a drop down selection list. This lists the available setups and sweeps. As a minimum, the LastAdaptive solution and AdaptivePass solution is available to choose.

     The AdaptivePass solution context can be selected to allow any value or parameter to be plotted versus the adaptive pass. This function is usually used to evaluate the convergence of the solution.
   - *For Transient projects only*:

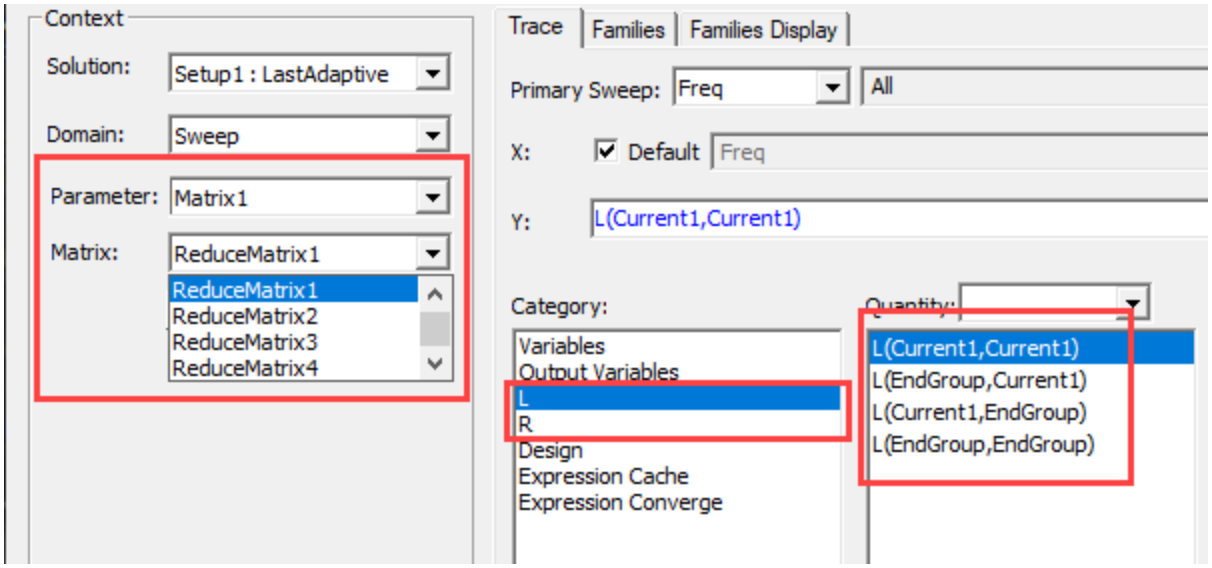
     Domain field with a pull down selection list containing options for plotting vs time (Sweep Domain) or plotting vs frequency (Spectral Domain).
   - *For Transient projects only*:

     Domain field with a pull down selection list containing options for **Average and RMS** and **Transient D-Q** for electric machines. Click **Machine Options** to setup the machine specific parameters.
   - *For Eddy Current projects only*:

     Domain field with a pull down selection list containing options for plotting vs frequency (Sweep Domain) or plotting vs time (Time Domain).

     Matrix field with a pull down selection list containing options for plotting *matrix* and *reduce matrix* parameters.

Note	In Maxwell Eddy Current designs, the user can create *matrix* parameters, which will cause the solver to produce an impedance *matrix* for the selected excitations. In addition, the user can group (wire) two or more excitations to one excitation in either a series or parallel connection referred to as a *reduce matrix*. The **Matrix** field appears only if a matrix entry is selected in the **Parameter** field. (Refer to *Assigning a Matrix* for information on creating *matrix* and *reduced* matrix parameters for Eddy Current designs.)



- **Parameter** field with a drop down selection list. Whether this field appears, and the parameters listed depend on the Solution type and the ***<type>*** selected.
- *For Fields and Noise Vibration reports*: **Geometry** field with a drop down selection list. This applies the quantity to a specific geometry.

6. The **Families** tab provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined (for example, for a parametric sweep). If so, the variables other than the one chosen as the **X (Primary sweep)**, are listed under the **Families** tab with columns for the variable, the value, and an Edit column with an ellipsis **[...]** button. See Using Families tab for Reports.
7. In the **Y** Component section of the dialog make selections for the following:
   a. Categories - those depend on the Solution type and the design. For example, Magnetostatic categories include: Torque, Output Variables, Inductance, and other user-selectable solution parameters. Transient categories include: Loss, Output Variables, Variables, and others. The selected category provides the default name of the plot, for instance Force Plot *n*. You can edit the plot names in the project tree and the plot header text in the report synchronizes.
   b. Quantities for Y are relative to the selected category. For example, Loss quantities include: CoreLoss, EddyCurrentLoss, HysteresisLoss, and others.

Note	The Quantity text field can be used to filter the Quantity list by typing in text, or by using the four predefined selections. This is useful if the Category selected produces a lengthy Quantities list. See Filtering Quantity Selections for the Reporter. When the matrix is very large, the number of quantities can be correspondingly huge. Therefore, the Quantities field can optionally use a tree structure to divide matrix quantities into groups by their first element name. The initial display shows groups, without initially listing group members.

c. **Function** list to apply to the Y quantities.

d. Value field displays the currently specified Quantity and Function. You can edit this field directly.

Note	Color shows valid expression.

e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.

8. In the **X (Primary Sweep)** section, make selections for the following:

a. Select the Primary sweep from the drop down menu. By default **All** of the chosen sweep's values are used. You can also select the browse [...] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time sweeps), or **Use all values** (the default setting).

With Use all values unchecked, you can select one or more by clicking an individual value, dragging to select multiple values, or using Alt-Click to specify specific values.

You select either the Sweep radio button for Default or Edited selection.

You can also select the browse [...] button here to display the Edit Sweep dialog for Modify Reports, which includes additional editing features.

To select an X component that is different from the Primary Sweep, uncheck the Default field to enable the X field and browse [...] button. Click the browse [...] button to display the **Select X Component** dialog box. This lets you specify the X component as you do the Y; that is, in terms of Categories which define the selectable Quantities, and Functions to apply. After making selections, **OK** the dialog to assign the X component.

b. The **Families** tab provides a way to select from valid solutions for sweeps where a simulation has multiple variables defined (for example, for a parametric sweep). If so, the variables other than the one chosen as the **X (Primary sweep)**, are listed under the **Families** tab with columns for the variable, the value, and an Edit column with an ellipsis [**...**] button. See Using Families tab for Reports.

You can also view and edit the properties of Reports and their traces via their Properties windows. See Modifying the Background Properties of a Report.

You can also modify the display type of an existing plot from the Properties dialog for that plot. Select the Report icon in the **Project** tree to display the **Properties** dialog box. Selecting the Display Type field displays a menu with selections available for that plot. Once you make a selection, the plot display updates for the current selection.

Note	Remember that for many excitations of interest for plotting, you can control the default base names through the dialog described here: Setting Default Boundary/Excitation Base Names. This may save you the need to edit individual names in the plots.

Related Topics

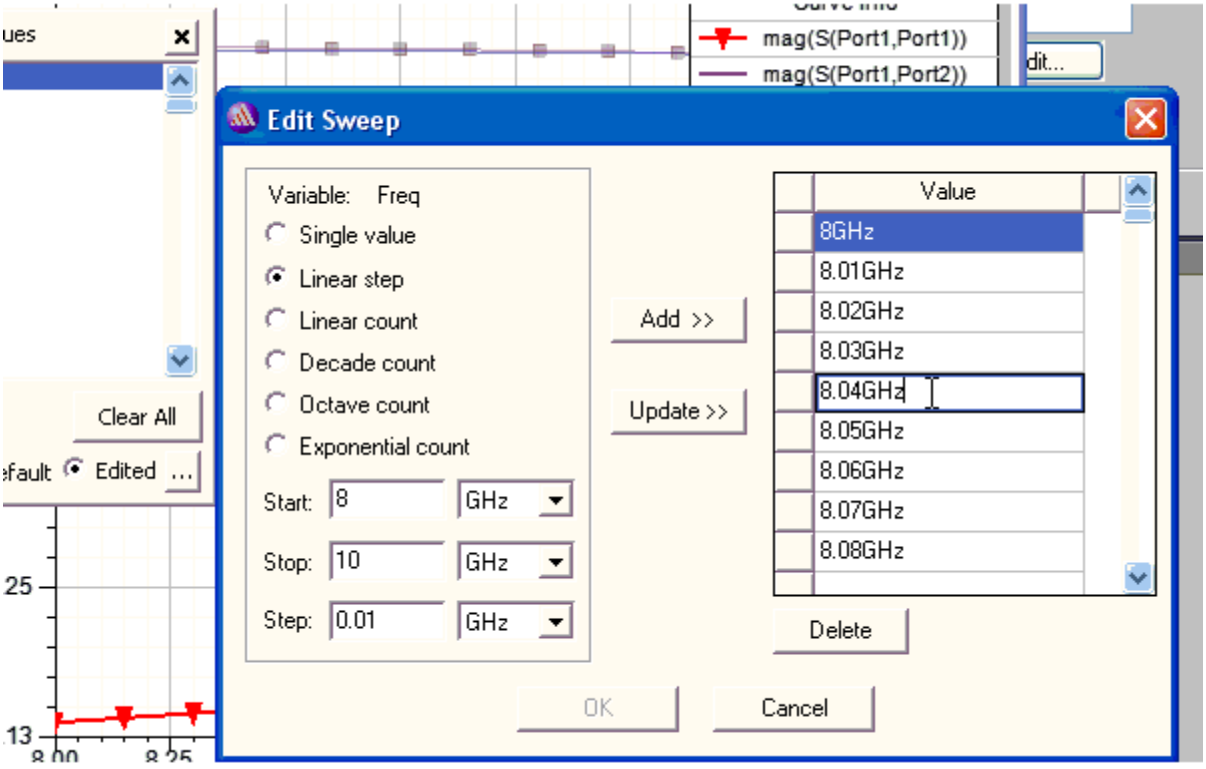
Modifying the Background Properties of a Report

Modifying the Legend in a Report

Creating Custom Report Templates
Working with Traces
Editing the Display Properties of Traces
Setting Default Boundary/Excitation Base Names
Discarding Report Values Below a Specified Threshold
Adding Trace Characteristics
Adding Data Markers to Traces

Modify Report: Using the Edit Sweep Dialog

Clicking the browse [...] button at the lower right corner of the Use all values pane opens the Edit Sweep dialog for Modify Report. The lets you edit the current Primary sweep variable values, including radio button selections for Single value, Linear step or count, and Decade, Octave, or Exponential Counts.



You can specify start, stop and step values and units, and add specific values to the list of current sweep values. The **Add>>** and **Update>>** buttons let you edit the value list.

You can use the mouse click, drag, and Alt-Click to select values. You can also edit individual values.

Related Topics

Creating Reports
Modifying Reports

Zooming and Fitting Reports

The standard Zoom and Fit commands operate on reports. After clicking in an open report, you can also use a mouse wheel, to zoom in and out.

Related Topics

Modifying the Background Properties of a Report

Modifying the Background Properties of a Report

To modify the appearance of a report, or the display properties an object in a report:

1. Open the report you want to modify.
2. You must select an editable object in the report to be able to edit its properties. Click on an object to select it and to view its Properties in the docked properties window. To open a floating Properties window, either double click on the selected object, or select **Edit>Properties**.

   The Properties tabs and options displayed for editable plot objects varies depending on the report type (for example, whether 2D rectangular, Stacked, or 3D), and can include the following:

   - **Cartesian** -- this lets you edit the scrollbar and thumb properties for 2D rectangular plots.
   - **Header** – this lets you edit the Properties for the text displayed at the top of the report, including the Title font, Company Name, Show Design Name, Subtitle Font. The plot title is tied to the report's name and is not a Header property. If you change the report name in the Project tree, plot title synchronizes. The Company Name and the Show Design Name check box are grouped in the Properties dialog as Subtitle. Edits to the Subtitle Font Property affects both of them.
   - **General** – this dialog (or General tab for other Report properties windows) lets you edit the background color (the perimeter around the trace display) for the plot, the contrast color (the trace display background), the Field width, the Precision, the Visual Detail level, and whether to use scientific notation for marker and delta marker displays. (X and Y notation display is set separately, in the Axis property tabs.) An Auto Scale property on by default and when enabled scales text in plots and colorkey (contour plot, field plots in 3D modeler) for high resolution screens.
   - **Legend --** this lets you edit the Properties for whether to Show Trace Name, Solution Name, and Variation Key. At least one of these three must be selected. You can also edit the Font, the background color of the Legend box, the Border Color, the Border Width, Grid Color (for the lines between Trace descriptions), and the Grid line width. Also see Modifying the Legend in a Report

- **Color Key** -- for 3D plots, to control the appearance of the color key (colors, transparency, border appearance, fonts, number format, Min and Max, field width and precision.
- **Contour** -- for 3D plots, to control the appearance of the color map, including map type, ramp color, spectrum, IsoValType, levels, number of contours, and values shown.
- **Stacked** -- for stacked plots, properties for X scrollbar, thumb properties, and stack layout, auto fit, and stack height.
- **Traces** – you can select traces either in the Legend or on the plot. The properties for traces include: Color, Line Style, Line Width, Trace Type, whether to Show a symbol, Symbol Frequency, Symbol style, whether to Fill symbol, symbol color, and whether to Show arrows. See Editing the Display Properties of Traces.
- X, Y, or Z **Axis**-- the defaults for most of these values are set in the **Report 2D Options** Axis tab.
  - Display name -- check box for whether to display the axis name.
  - Specify name -- check box for specifying the Axis name.
  - Name -- this describes the axis to which the following properties/options refer. These are selected in the Report dialog.
  - Axis Color -- set the color by double clicking to display the Set color dialog box. Select a default or custom color and click **OK**.
  - Axis Font -- click the cell to display the Edit Text Font dialog box. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. OK the selections to apply the font edits and to close the dialog.
  - Show Units -- this specifies whether to display units.
  - **Window (section)**
    - Window Mode-- can be Axis range, Continuous moving window, or Step moving window.
    - Window Width (in) -- provide an integer value for the previous selection.
  - **Manual Format** (section)
    - Number format -- select from the drop down menu, Auto, Decimal, or Scientific notation.
    - Field Width -- enter a real value.
    - Field Precision -- enter a real value.
- X, Y, or Z **ScalingTab** – These properties provide control over scaling.
  - Axis Scaling -- use the drop down menu to select scaling as Linear or Log. For the Y axis, all zero or negative values are discarded before log scaling is applied.
  - Specify Min -- check box
  - Min -- text entry in same units as axis units. Saved as SI internally.
  - Specify Max -- check box
  - Max -- text entry in same units as axis units. Saved as SI internally.
  - Specify Spacing -- check box
  - Spacing -- text entry in same units as axis units. Saved as SI internally

- **Manual Units** (section)
  - Auto Units -- use the check box compute the correct units for the axis.
  - Units -- click on the cell to select from a menu of available units if you have not checked Auto Units.
- **Infinity Visualization** (section)
  - Map Infinity Mode -- check box.

  Each axis now can be set to treat infinity values in a user defined way. When you check the Map Infinity Mode, any infinity values in the input data get the infinityMap value (negative infinity get the value*-1 and positive infinity the positive value specified). This can be useful if there are zeros, or very small values that can be treated as zero, in the data, for example, dB Gain.
  - Map Infinity To -- enter a real value for the Map Infinity Mode.
- **Grid** -- properties for grid labels and grid style, appearance, line styles, color, major and minor lines, and scaling. For the 3D rectangular plots, there are separate tabs for the XY, YZ and ZX axes.

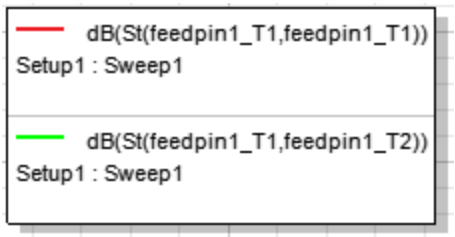
3. Edit the properties, and click **OK** to apply the changes.

Related Topics

Modifying Reports
Working with Traces
Discarding Report Values Below a Specified Threshold
Modifying the Legend in a Report
Editing the Display Properties of Traces
Creating Custom Report Templates and Defaults
Setting Report2D options

Modifying the Legend in a Report

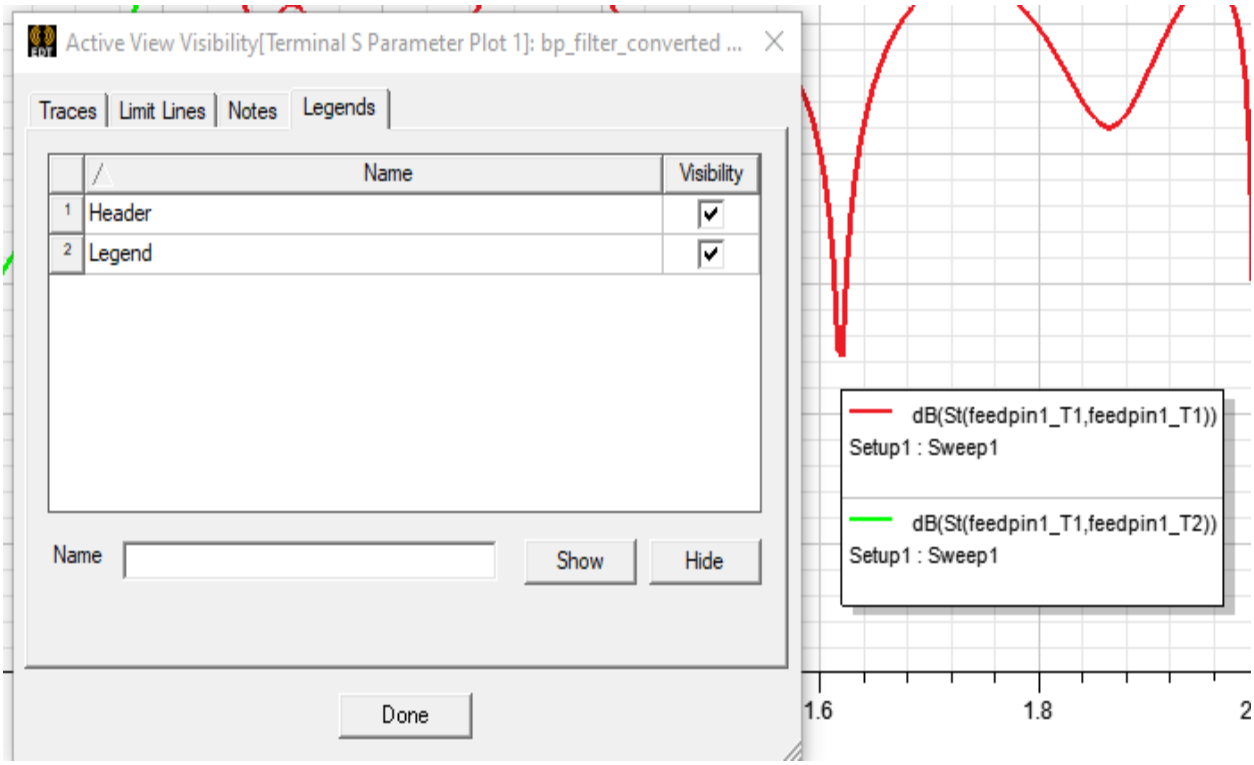
The legend in a report is a list of the curves being plotted. For each curve, the legend gives the name, shows the line color, and lists the setup and the adaptive pass used to generate the curve.



To **show** or **hide** a legend in a report:

1. Make the report the active view.
2. Use **View>Active View Visibility** display or hide the report and display the **Active View Visibility** dialog.
3. Select the **Legends** tab.

   This lists the legend (or legends) in the report.

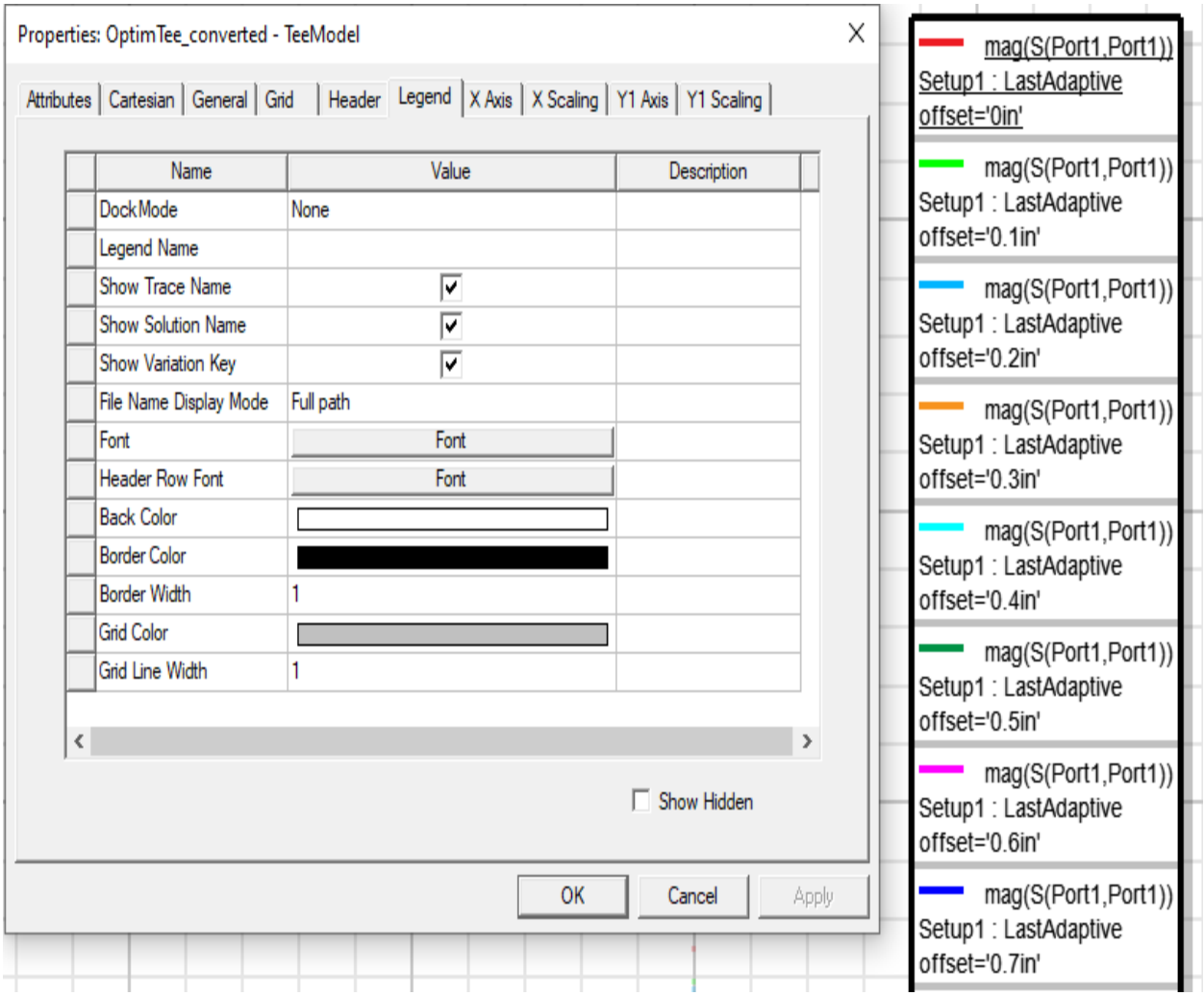


4. Check the visibility check box, and OK the dialog to close it and apply the change.

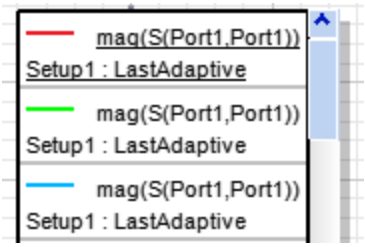
To **edit the display properties** of a legend:

1. Select the legend in a report by clicking on the Curve Info panel to display a docked properties window, or right-click on the legend and select **Edit>Properties** to display the floating properties window.

   This lets you edit the Properties for Dock Mode, Legend Name (default is no name. When non-empty, a header row for the Legend in plot shows up with that string.), whether to Show Trace Name, Solution Name, and Variation Key (which applies to parametric variables, if present). If none of these three are selected, only a trace color shows.



Here is an example with Variation Key off.



You can also edit the Font by clicking the Font cell to display the **Edit Text Font** dialog box. The dialog lets you select from a list of available fonts, styles, sizes, effects, colors, and script. The dialog also contains a preview field. OK the selections to apply the font edits and to close the dialog box.

You can also edit the background color of the Legend box, the Border Color, the Border Width, Grid Color (for the lines between Trace descriptions), and the Grid line width.

2. Click **OK** to close the Properties window and apply the selections.

To **change the display name** for traces, see Editing Trace Properties.

To **move** a legend in a report:

1. Click and hold on the legend.
   The cursor changes to crossed lines with arrow tips.
2. Still holding, drag the legend to a new location and release.
   The legend is released and the crossed lines change back to a mouse pointer.

To **resize** a legend in a report:

1. Position the mouse tip over the edge you want to resize.
   The mouse pointer changes to a horizontal or vertical line with arrow tips.
2. Click and drag the horizontal or vertical edge to the desire size.
3. Release.

Related Topics

Editing Trace Properties
Showing Objects
Hiding Objects from View
Modifying Reports
Creating Custom Report Templates
Discarding Report Values Below a Specified Threshold
Setting Report2D options
Editing the Display Properties of Traces

Selecting the Report Type

The **Report Types** available for creating a report depends on the simulation setup. The Report Category provides the default report name for the project tree and the text displayed in the report header. For example, Force as the Category type causes the default report name to be Force Plot *n*. Depending on the design setup, you can make a selection from the following report types.

Select from the following report types:

Magnetostatic	Project variables, user specified Output Variables and Design data are available to plot. Depending upon the solution parameters requested, torque data may be available to plot, as well as Inductance, and magnetic flux. Quantities such as Lnom, and MagFluxNom may also be available for plotting. See Post processed Quantities for more details.
Electrostatic	Project variables, user specified Output Variables and Design data are available to plot. Depending upon the solution parameters requested, torque data may be available to plot, as well as Capacitance.

Eddy Current	Project variables, user specified Output Variables and Design data are available to plot. Depending upon the solution parameters requested, torque data may be available to plot, as well as Inductance, Resistance, Impedance. Quantities such as Lnom, MagFlux, and various Winding, Loss, and End Connection quantities may also be available for plotting, depending on the design. See Post processed Quantities for more details.
DC Conduction	Project variables, Output Variables, and Design data are available for plotting, as well as Resistance matrix data.
AC Conduction (2D Only)	Project variables, Output Variables, and Design data are available for plotting. Depending upon the solution parameters requested, Capacitance, Conductance, and Admittance quantities may also be available for plotting.
Transient	Project variables, Output Variables, and Design data are available for plotting, as well as Force, Torque, Loss, Winding, and solution process data.
Transient A-Phi Formulation (3D only)	Project variables, Output Variables, and Design data are available for plotting, as well as Force, Torque, Loss, Winding, and solution process data. Terminal Voltage, Terminal Current, Inductance, and Capacitance quantities may also be available for plotting, depending on the design.
Electric Transient (3D Only)	Project variables, Output Variables, and Design data are available for plotting, as well as Force, Torque, Loss, and solution process data.
Fields	Project variables, Output Variables, and Design data are available for plotting, as well as calculator expressions.
Noise Vibration	Project variables, Output Variables, and Design data are available for plotting, as well as calculator expressions.
Harmonic Force	Project variables and Output Variables, are available for plotting, as well as harmonic force data (radial, tangential, and axial complex values of harmonic force are available for transient 3D designs – only radial and tangential for transient 2D designs).

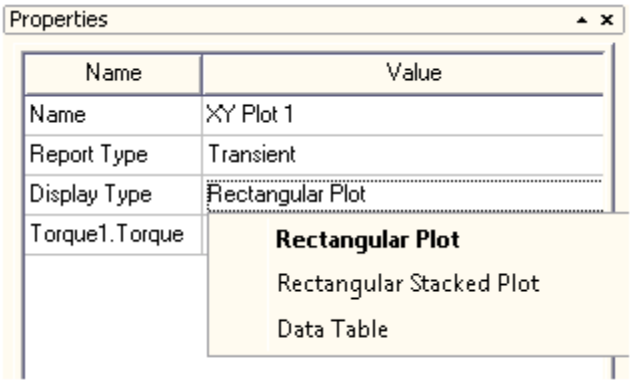
Selecting the Display Type

The information in a report can be displayed in several formats. Select from the following **Display Type** formats in the **Create Report**dialog box:

Rectangular Plot	A 2D rectangular (x-y) graph.
Rectangular Stacked Plot	A series of 2D rectangular (x-y) graphs stacked vertically, sharing a common x-axis scale, but with each trace having its own y-axis scale.
Data Table	A spreadsheet with rows and columns that displays, in numeric form, selected quantities against a swept variable or another quantity.
3D Rectangular	A 3D rectangular (x-y-z) graph.

Plot	
3D Rectangular Bar Plot	A 3D rectangular (x-y-z) bar graph.
Rectangular Contour Plot	A rectangular (x-y-z) graph. Contour plots are useful to visualize surfaces (for e.g. Directivity as a function of phi/theta).

You can also modify the display type of an existing plot from the **Properties** window for that plot. Select the Report icon in the Project tree to display the Properties window. Selecting the **Display Type** field displays a menu with selections available for that plot.



Once you make a selection, the plot display updates for the current selection.

Creating 2D Rectangular Plots

A rectangular plot is a 2D, x-y graph of results.

1. On the **Results** menu (Maxwell menu or right-click on **Results** on the Project tree), click **Create <*type*> Report**, and select **Rectangular Plot**.

   The **Report** dialog appears.
2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   - **Solution** field with a drop down selection list. This lists the available setups and sweeps. As a minimum, the LastAdaptive solution and AdaptivePass solution is available to choose.

     The AdaptivePass solution context can be selected to allow any value or parameter to be plotted versus the adaptive pass. This function is usually used to evaluate the convergence of the solution.
   - *For Transient projects only*:

     Domain field with a pull down selection list containing options for plotting vs time (Sweep Domain) or plotting vs frequency (Spectral Domain).

- *For Transient projects only*:

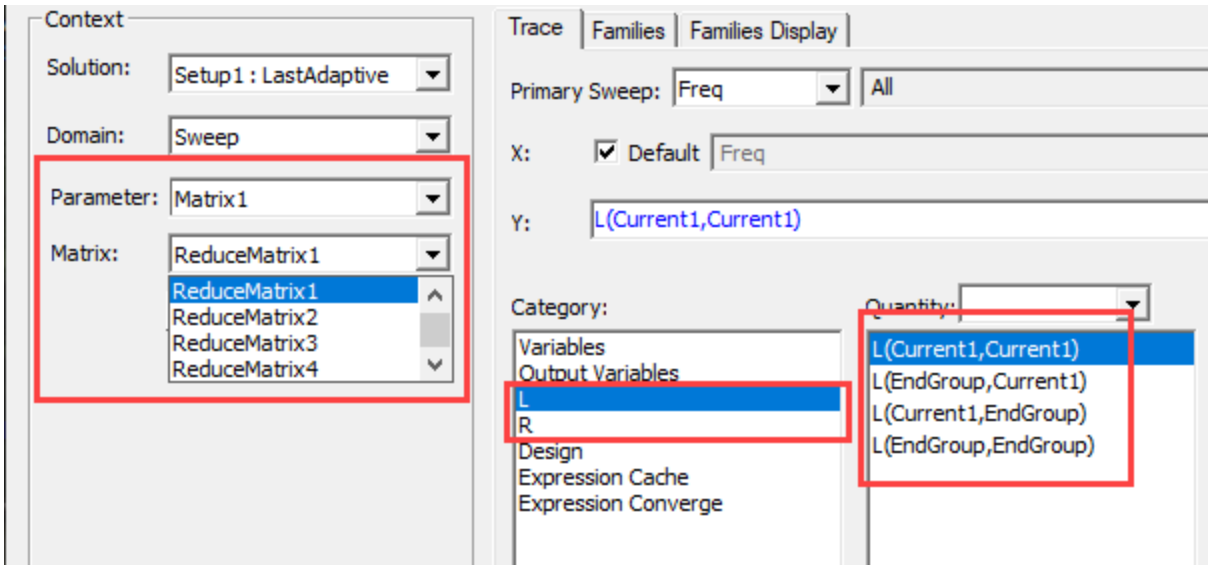
  Domain field with a pull down selection list containing options for **Average and RMS** and **Transient D-Q** for electric machines. Click **Machine Options** to setup the machine specific parameters.

- *For Eddy Current projects only*:

  Domain field with a pull down selection list containing options for plotting vs frequency (Sweep Domain) or plotting vs time (Time Domain).

  Matrix field with a pull down selection list containing options for plotting *matrix* and *reduce matrix* parameters.

Note	In Maxwell Eddy Current designs, the user can create *matrix* parameters, which will cause the solver to produce an impedance *matrix* for the selected excitations. In addition, the user can group (wire) two or more excitations to one excitation in either a series or parallel connection referred to as a *reduce matrix*. The **Matrix** field appears only if a matrix entry is selected in the **Parameter** field. (Refer to *Assigning a Matrix* for information on creating a *matrix* and *Assigning a Reduce Matrix* for information on creating *reduced* matrix parameters for Eddy Current designs.)



- **Parameter** field with a drop down selection list. Whether this field appears, and the parameters listed depend on the Solution type and the ***<type>*** selected.
- *For Fields and Noise Vibration reports*: **Geometry** field with a drop down selection list. This applies the quantity to a specific geometry.

3. Under the **Trace** tab, **Y** component section, specify the information to plot along the y-axis:
   a. In the **Category** list, click the type of information to plot.
   b. In the **Quantity** list, click the value to plot.
   c. In the **Function** list, click the mathematical function of the quantity to plot.
   d. Value field displays the currently specified Quantity and Function. You can edit this field directly.

Note	Color shows valid expression.

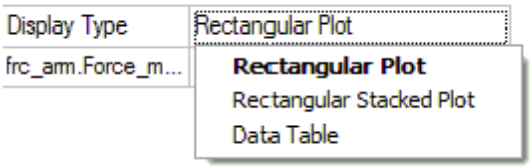
e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.

4. On the **Trace** tab, **X** (Primary sweep) line, specify the quantity to plot along the x-axis in one of the following ways:
   - Select the sweep variable to use from the drop down list.
   - If sweeps are available, you can select the browse [...] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time sweeps), or **Use all values** (the default setting). The quantity will be plotted against the primary sweep variable listed.
5. On the **Families** tab, confirm or modify the sweep variables that will be plotted.
6. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog box. The default name is based on the Report Category you selected, (for example, Force Plot *n* or Output Variables Plot *n*). You can edit the plot names in the project tree and the plot header text in the report synchronizes.

   The function of the selected quantity will be plotted against the swept variable values or quantities you specified on an x-y graph. The plot is listed under **Results** in the project tree and the traces are listed under the plot. When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.
7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace rather than New Report**.

You can also modify the display type of an existing plot from the Properties dialog for that plot. Select the Report icon in the Project tree to display the Properties dialog box. Selecting the Display Type field displays a menu with selections available for that plot.



Once you make a selection, the plot display updates for the current selection.

Related Topics

Sweeping a Variable

Working with Traces

Adding Trace Characteristics

Delta Markers in 2DPlots
Modifying Background Properties of a Report
Discarding Report Values Below a Specified Threshold
Setting Report2D options
Plotting a Quantity versus Adaptive Pass

Creating a 2D Rectangular Stacked Plot

A rectangular stacked plot is a 2D, x-y graph of results, with each trace displayed on a separate plot.

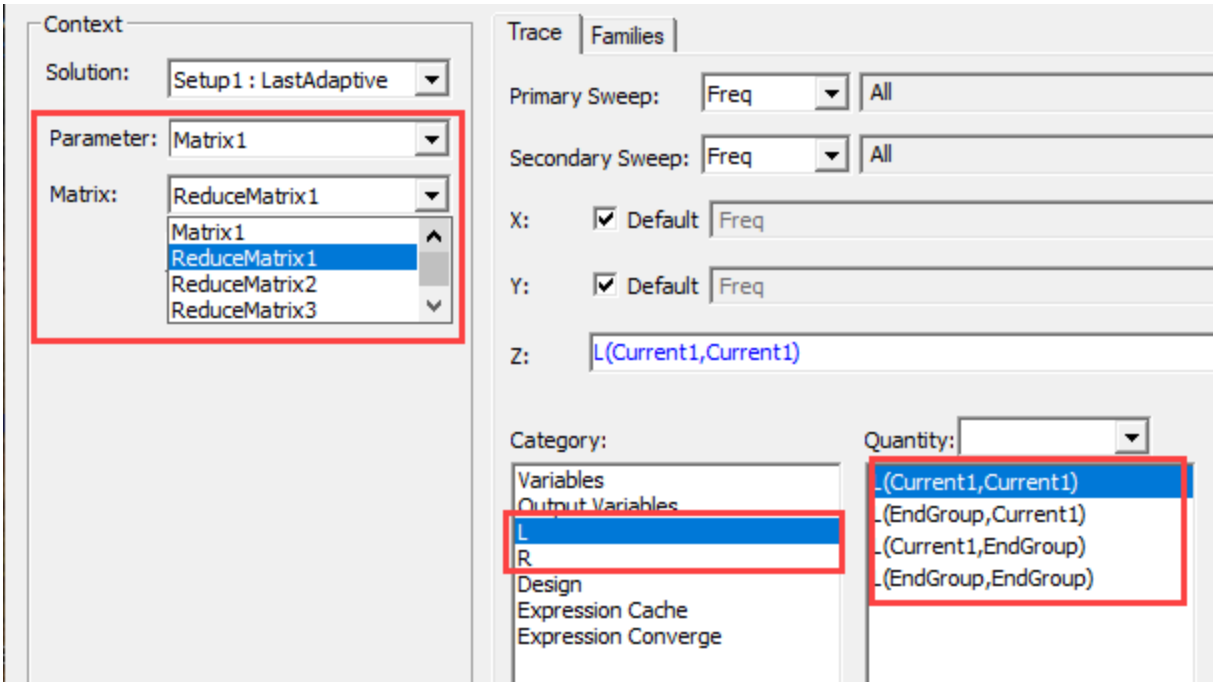
1. On the **Results** menu (Maxwell menu or right-click on **Results** on the Project tree), click **Create <*type*> Report**, and select **Rectangular Stacked Plot**.

   The **Report** dialog appears.
2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   - **Solution** field with a drop down selection list. This lists the available setups and sweeps. As a minimum, the LastAdaptive solution and AdaptivePass solution is available to choose.

     The AdaptivePass solution context can be selected to allow any value or parameter to be plotted versus the adaptive pass. This function is usually used to evaluate the convergence of the solution.
   - **Parameter** field with a drop down selection list. Whether this field appears, and the parameters listed depend on the Solution type and the ***<type>*** selected.
   - *For Eddy Current projects only*:

     Matrix field with a pull down selection list containing options for plotting *matrix* and *reduce matrix* parameters.

Note	In Maxwell Eddy Current designs, the user can create *matrix* parameters, which will cause the solver to produce an impedance *matrix* for the selected excitations. In addition, the user can group (wire) two or more excitations to one excitation in either a series or parallel connection referred to as a *reduce matrix*. The **Matrix** field appears only if a matrix entry is selected in the **Parameter** field. (Refer to *Assigning a Matrix* for information on creating a *matrix* and *Assigning a Reduce Matrix* for information on creating *reduced* matrix parameters for Eddy Current designs.)



- *For Fields and Noise Vibration reports*: **Geometry** field with a drop down selection list. This applies the quantity to a specific geometry.

3. Under the **Trace** tab, **Y** component section, specify the information to plot along the y-axis:
   a. In the **Category** list, click the type of information to plot.
   b. In the **Quantity** list, click the value to plot.
   c. In the **Function** list, click the mathematical function of the quantity to plot.
   d. Value field displays the currently specified Quantity and Function. You can edit this field directly.

Note	Color shows valid expression.

   e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.
4. On the **Trace** tab, **X** (Primary sweep) line, specify the quantity to plot along the x-axis in one of the following ways:
   - Select the sweep variable to use from the drop down list.
   - If sweeps are available, you can also select the browse [...] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time sweeps), or **Use all values** (the default setting). The quantity will be plotted against the primary sweep variable listed.
5. On the **Families** tab, confirm or modify the sweep variables that will be plotted.
6. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog box. The default name is based on the

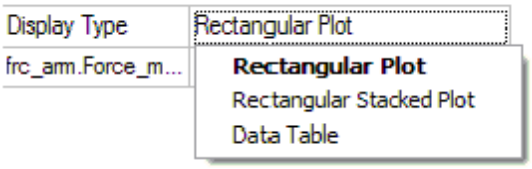
Report Category you selected, (for example, Force Plot *n* or Output Variables Plot *n*). You can edit the plot names in the project tree and the plot header text in the report synchronizes.

The function of the selected quantity will be plotted against the swept variable values or quantities you specified on an x-y graph. The plot is listed under **Results** in the project tree and the traces are listed under the plot. When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.

7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

   The new trace will appear as a separate plot “stacked” above the first plot. The stacked plots share a common x-axis scale. However, each plot has its own independent y-axis scale.

You can also modify the display type of an existing plot from the Properties window for that plot. Select the Report icon in the Project tree to display the Properties window. Selecting the Display Type field displays a menu with selections available for that plot.



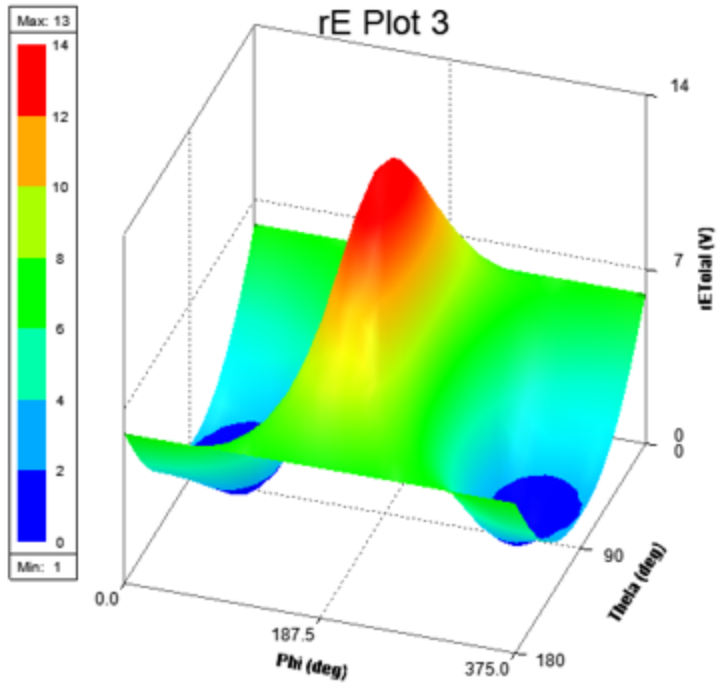
Once you make a selection, the plot display updates for the current selection.

Related Topics

Sweeping a Variable
Working with Traces
Delta Markers in 2DPlots
Modifying Background Properties of a Report
Plotting a Quantity versus Adaptive Pass
Adding Data Markers to Traces

Creating 3D Rectangular Plots

Below is an example of a 3D rectangular plot.



Working with a 3D Rectangular Plot

You can Rotate, Zoom and Pan a plot. When you rotate, the Cartesian grid responds so that the curve always remains in front and the grids behind.

Clicking on a plot entity selects it, highlighting the selected entity in bold.

Double-clicking anywhere in the plot brings up the Properties dialog box. the properties are grouped appropriately under various tabs, which correspond to plot entities:

- General: For general plot properties such as Visual Detail level and background color
- Header: Properties related to plot Header/Title.
- Axis [X|Y|Z]: Properties related to the 3 axes
- Grid [XY|YZ|ZX]: Properties related to the 3 grids
- ColorKey: Properties related to ColorKey, including borders, background, Min and Max, as well as number format and precision.
- Contour: Properties related to contouring of all curves/surfaces
- Surface: Properties related to the curve

Selecting a property also displays its properties in the Property window. You can edit the properties to customize the appearance of the plot. See "Controlling Visual Detail in a 3D Plot" on page 23-56 .

Creating a 3D Rectangular Plot

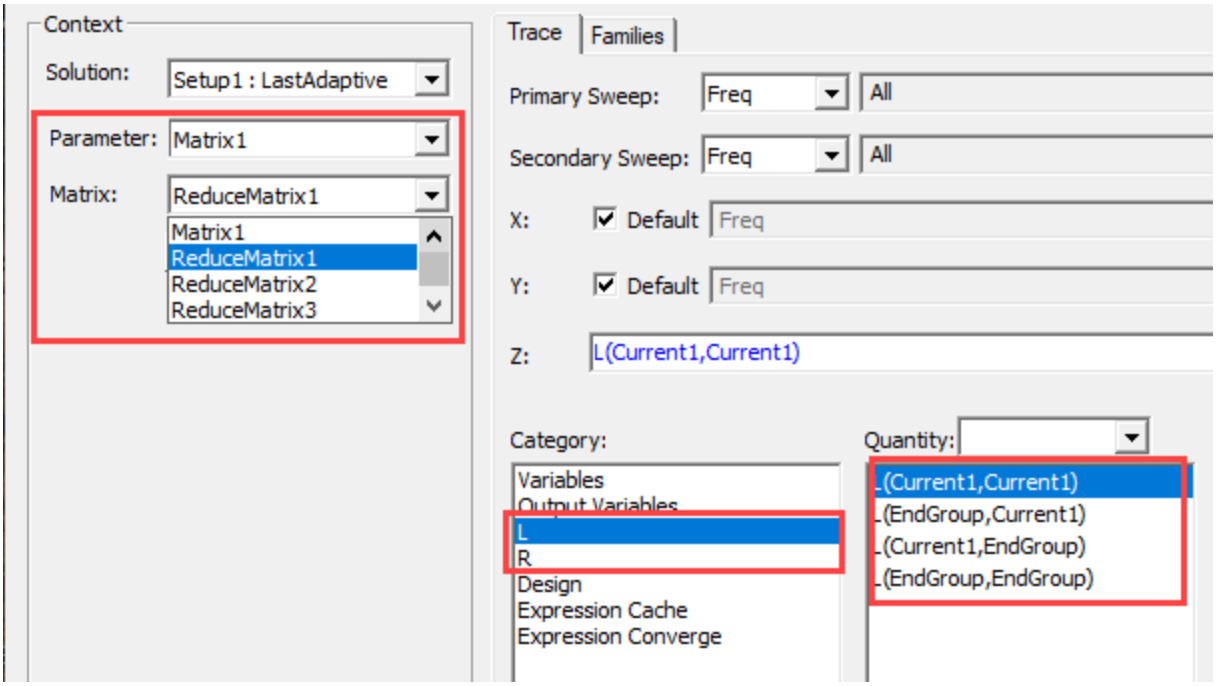
1. On the **Results** menu (Maxwell or RMxprt menu or right-click on **Results** on the Project tree), click **Create <*type*> Report**, and select **3D Rectangular Plot** from the report type menu.

   The **Report** dialog appears.
2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   - **Solution** field with a drop down selection list. This lists the available setups and sweeps. As a minimum, the LastAdaptive solution and AdaptivePass solution is available to choose.

     The AdaptivePass solution context can be selected to allow any value or parameter to be plotted versus the adaptive pass. This function is usually used to evaluate the convergence of the solution.
   - **Parameter** field with a drop down selection list. Whether this field appears, and the parameters listed depend on the Solution type and the **<*type*>** selected.
   - *For Eddy Current projects only*:

     Matrix field with a pull down selection list containing options for plotting *matrix* and *reduce matrix* parameters.

Note	In Maxwell Eddy Current designs, the user can create *matrix* parameters, which will cause the solver to produce an impedance *matrix* for the selected excitations. In addition, the user can group (wire) two or more excitations to one excitation in either a series or parallel connection referred to as a *reduce matrix*. The **Matrix** field appears only if a matrix entry is selected in the **Parameter** field. (Refer to *Assigning a Matrix* for information on creating a *matrix* and *Assigning a Reduce Matrix* for information on creating *reduced* matrix parameters for Eddy Current designs.)



- *For Fields and Noise Vibration reports*: **Geometry** field with a drop down selection list. This applies the quantity to a specific geometry.
- For Rmxprt only: Domain field with a pull down selection list containing options for plotting vs ElectricDegree, Speed, or a user-selected Parameter.

3. Under the **Trace** tab, **Z** Component area, specify the information to plot along the z-axis:
   a. In the **Category** list, click the type of information to plot.
   b. In the **Quantity** list, click the value to plot.
   c. In the **Function** list, click the mathematical function of the quantity to plot.
   d. The **Value** field displays the currently specified Quantity and Function. You can edit this field directly.

Note	Color shows valid expression.

   e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.
4. On the **Trace** tab, **Y** (**Secondary Sweep**) lines, specify the information to plot along the y-axis in one of the following ways:
   - Select the sweep variable to use from the Secondary Sweep drop down list.
   - If sweeps are available, you can also select the browse [...] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time sweeps), or **Use all values** (the default setting). The quantity will be plotted against the primary sweep variable listed.
5. On the **Trace** tab, **X** (**Primary Sweep**) lines, specify the information to plot along the x-axis in one of the following ways:

- Select the sweep variable to use from the Primary Sweep drop down list.
- If sweeps are available, you can also select the browse [...] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time sweeps), or **Use all values** (the default setting). The quantity will be plotted against the primary sweep variable listed.

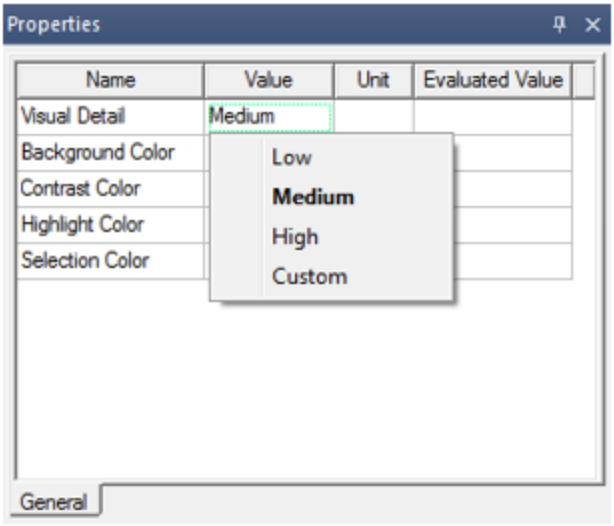
6. On the **Families** tab, confirm or modify the sweep variables that will be plotted.
7. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog box. The default name is based on the Report Category you selected, (for example, Force Plot *n* or Output Variables Plot *n*). You can edit the plot names in the project tree and the plot header text in the report synchronizes.

   The function of the selected quantity or quantities will be plotted against the values you specified on an x-y-z graph. The plot is listed under **Results** in the project tree.When you select the traces or plots, axis or grid labels, plot header, color key, or variable labels, their properties are displayed in the Properties window. The properties for each plot element can be edited directly to modify the plot content and appearance. See Modifying the Background Properties of a Report.
8. Optionally, add another trace to the plot by following the procedure above, using **Add Trace rather than New Report**.

Controlling Visual Detail in a 3D Plot

If a particular plot seems busy with information, you can edit plot properties, such as Axis and Grid Attributes for discrete levels of visual detail to improve readability. Double-click anywhere on a plot to display the Properties dialog box. The Visual Detail property on the General tab also provides control suited to different screen and plot sizes.

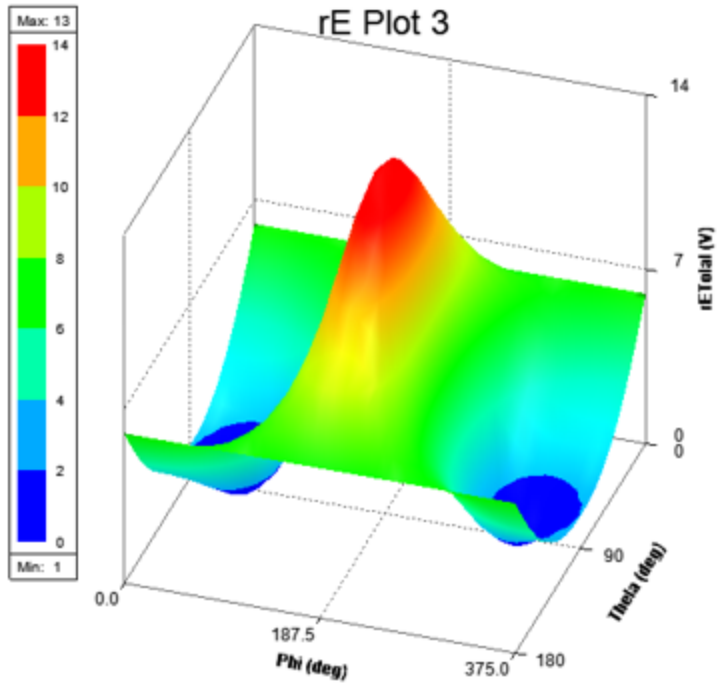


The Visual Detail menu has four options: Low, Medium (the default), High, and Custom. If you select any Visual Detail, the 3D plot is rendered according to the selected Visual Detail level and

the properties reflect the values chosen for the selected visual detail level. From this predefined visual detail level, if you modify any properties, Visual Detail is automatically set to Custom (or to another predefined visual detail level if the edits happen to match the settings for that level).

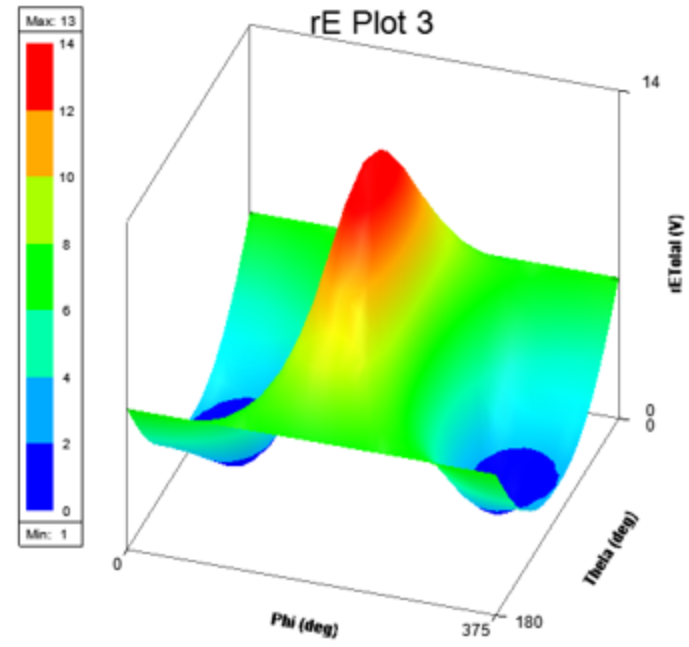
You can also manually set Visual Detail to Custom. In such a case, Custom will inherit property values corresponding to the previous level. This ensures that you can customize settings starting from a baseline provided by the preconfigured Low, Medium or High Visual Detail levels.

3D Rectangular Plot with Medium Visual Detail



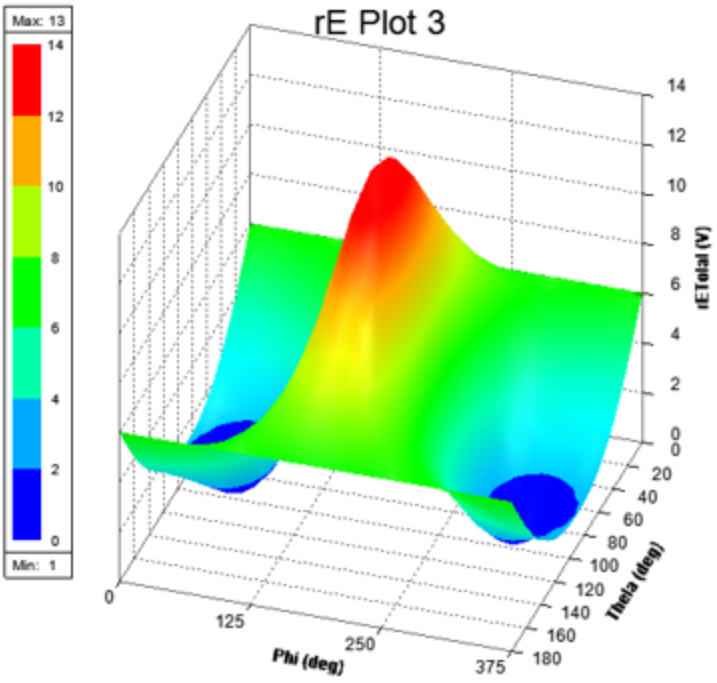
On creation, a 3D Rectangular Plot has Visual Detail set to Medium and looks and feels as shown above. Specifically, under Medium Visual Detail level, a 3D Rectangular Plot has 3 ticks per axis (X, Y, Z axis) which will show min, max and middle value. This setting also shows axes labels.

3D Rectangular Plot with Low Visual Detail



With Visual Detail set to Low, a 3D Rectangular Plot shows axes with 2 ticks corresponding to min and max values. It also shows axes labels and grid borders.

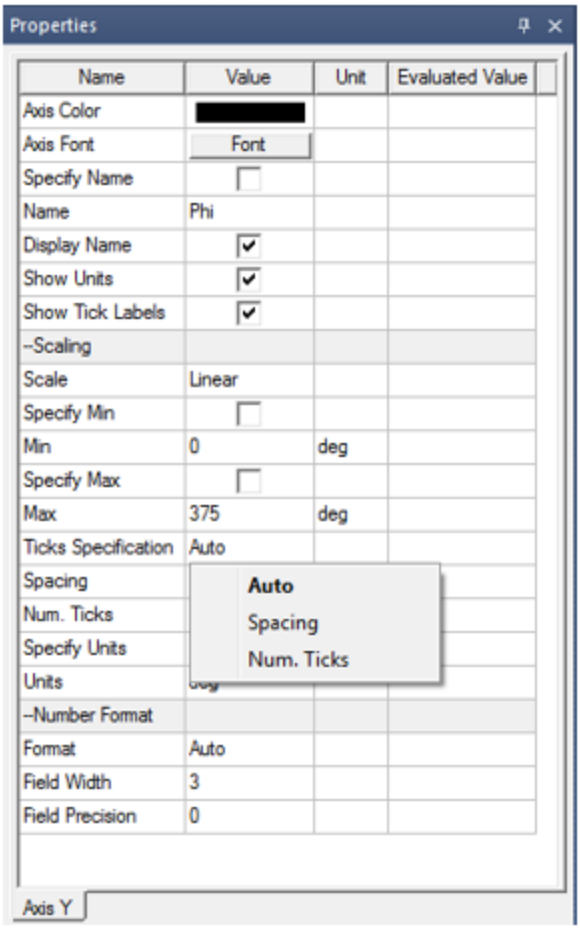
3D Rectangular Plot with High Visual Detail



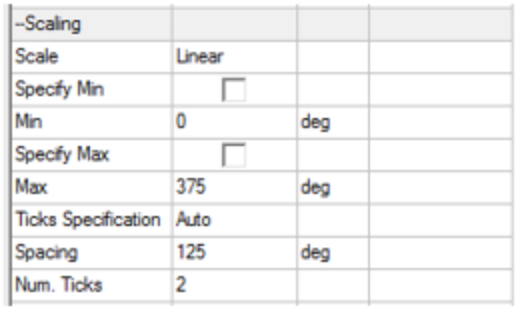
With Visual Detail set to High, a 3D Rectangular Plot shows all Cartesian axes and grids together with all ticks and axes labels.

Axis Properties: Ticks Specification and Num. Ticks

Ticks Specification is available on Axis properties, as shown below:



Ticks Specification is a menu with possible values as Auto, Spacing, and Num. Ticks, with Auto being the default value. If Ticks Specification is Auto, then a spacing value is automatically calculated and used to calculate and display the tick labels. **Spacing** shows the calculated value, and Num. Ticks shows the number of ticks based on this spacing value, as shown below:



You can edit the **Spacing** field when Ticks Specification is set to Spacing; otherwise, it is read only.

You can edit the **Num. Ticks** field when Ticks Specification is Num. Ticks; otherwise, it is read only.

Valid Num. Ticks are between 0 and 100, including 0 and 100. If you enter an invalid value, an error message is shown. If you enter a spacing value that results in number of ticks greater than 100, then an appropriate value is shown.

- If Num. Ticks is 0, then no ticks are shown on the axis.
- If Num. Ticks is 1, then only the max value tick is shown on the axis.
- If Num. Ticks is 2, then only the min and max value ticks are shown on the axis.
- If Num. Ticks is greater than 2, then evenly spaced ticks (including min and max) are shown on the axis.

Note	With the addition of the Ticks Specification property to Axis properties, the **Specify Spacing** property was removed as an Axis property. • If an R18.0 or R18.1 project is opened with **Specify Spacing** as Unchecked, Ticks Specification is set to Auto. • If an R18.0 or R18.1 project is opened with **Specify Spacing** as Checked, Ticks Specification is set to Spacing.

Related Topics

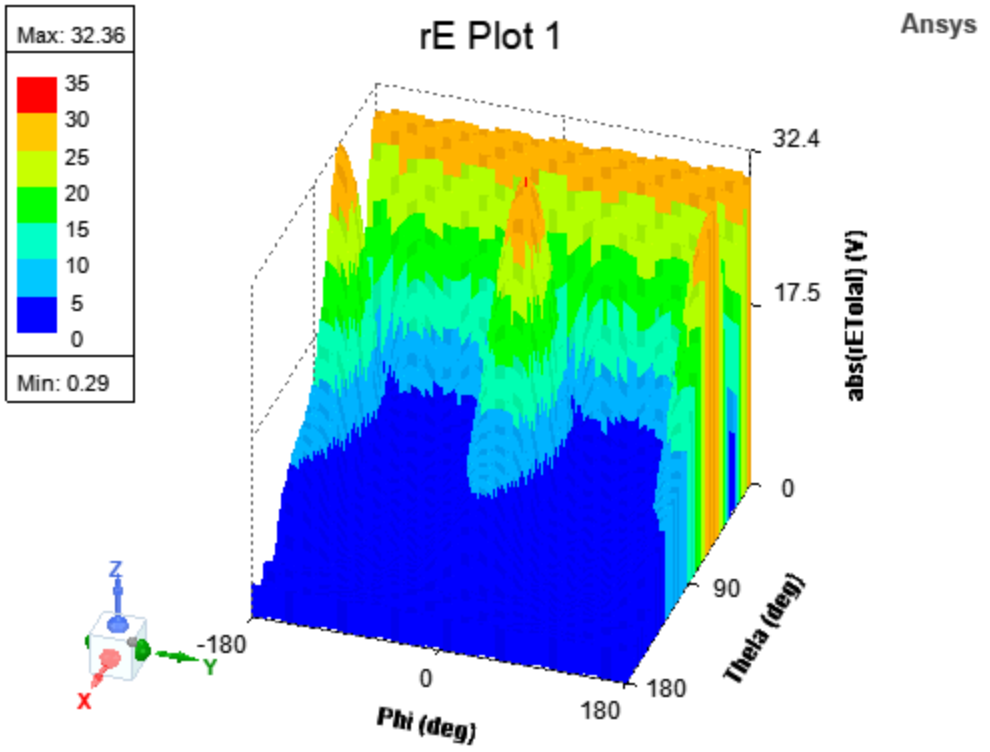
Sweeping a Variable

Working with Traces

Adding Trace Characteristics

Creating 3D Rectangular Bar Plots

This is a 3D, x-y-z graph of results as rectangular bars.

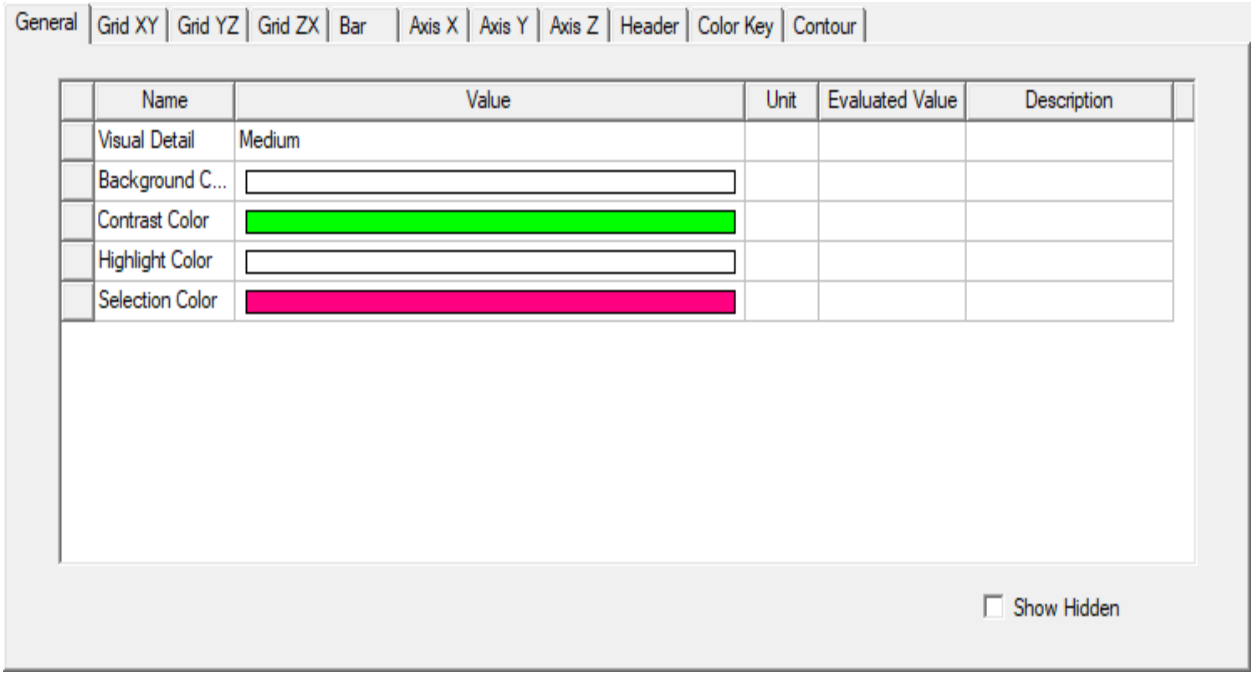


Working with a 3D Rectangular Plot

You can Rotate, Zoom, and Pan a plot. When you rotate, the Cartesian grid responds so that the curve always remains in front and the grids behind.

Clicking on a plot entity selects it, highlighting the selected entity in bold.

Double-clicking anywhere in the plot brings up the **Properties** dialog box.



The properties tabs for a 3D Rectangular Bar plot are:

- General: For general plot properties such as Visual Detail level and background color
- Grid [XY|YZ|ZX]: Properties related to the 3 grids
- Bar: Properties related to rectangular bars.
- Axis [X|Y|Z]: Properties related to the 3 axes
- Header: Properties related to plot Header/Title.
- ColorKey: Properties related to ColorKey, including borders, background, Min and Max, as well as number format and precision.
- Contour: Properties related to contouring of all surfaces

Selecting a property also displays its properties in the Property window. You can edit the properties to customize the appearance of the plot. See "Controlling Visual Detail in a 3D Rectangular Bar Plot" on page 23-66 .

Creating a 3D Rectangular Bar Plot

1. On the **Results** menu (under the **Maxwell** menu or right-click **Results** in the Project Manager), click **Create *<type>* Report**, and select **3D Rectangular Bar plot** from the report type menu, or select the Results tab of the ribbon, and select the 3D Rectangular Bar icon for the Report type you want to create.

   The *Report* dialog box appears.
2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   a. Solution field with a drop-down selection list. This lists the available solutions, whether sweeps or adaptive passes.

b. Geometry field with a drop-down selection list. For field and radiated field reports, this applies the quantity to a geometry or radiated field setup.

3. Under the **Trace** tab **Z** Component area, specify the information to plot along the z-axis:
   a. In the **Category** list, click the type of information to plot. The category you select provides the default plot name.
   b. In the **Quantity** list, click the value to plot.
   c. In the **Function** list, click the mathematical function of the quantity to plot.
   d. The **Value** field displays the currently specified Quantity and Function. You can edit this field directly.

      > **Note:**
      >
      > Color shows valid expression.

   e. **Range Function** button -- opens the *Set Range Function* dialog box. This applies currently specified Quantity and Function.
4. On the **Trace** tab **Y** (Secondary sweep) lines, specify the information to plot along the y-axis in one of the following ways:
   - Select the sweep variable to use from the drop-down list.
   - If sweeps are available, you can select the browse button to display a dialog that lets you select particular values. The quantity will be plotted against the primary sweep variable listed.
5. On the **Trace** tab **X** (Primary sweep) lines, specify the information to plot along the x-axis in one of the following ways:
   - Select the sweep variable to use from the drop-down list.
   - If sweeps are available, you can select the browse button to display a dialog that lets you select particular values. The quantity will be plotted against the primary sweep variable listed.
6. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables **Add Trace** on the **Report** dialog box. The default name is based on the Report Category you selected, (for example, S Parameter Plot *n* or rE Plot *n*). You can edit the plot names in the project tree and the plot header text in the report synchronizes.

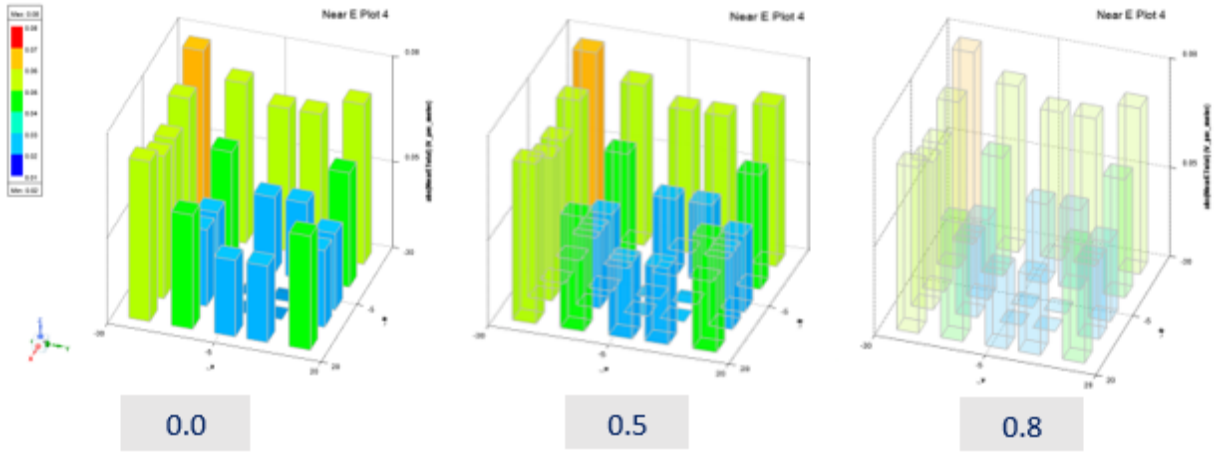
   The function of the selected quantity or quantities will be plotted against the values you specified on an x-y-z graph. The plot is listed under **Results** in the Project Manager. When you select the traces or plots, axis or grid labels, plot header, color key, or variable labels, their properties are displayed in the Properties window. The properties for each plot element can be edited directly to modify the plot content and appearance. See Modifying the Background Properties of a Report.
7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

3D Rectangular Bar Customization

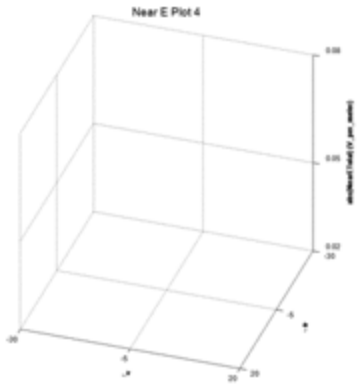
Selecting a bar in the plot shows the customizable attributes of bars in a 3D Rectangular Bar plot.

The customizable attributes of the bars in a 3D Rectangular Bar plot include:

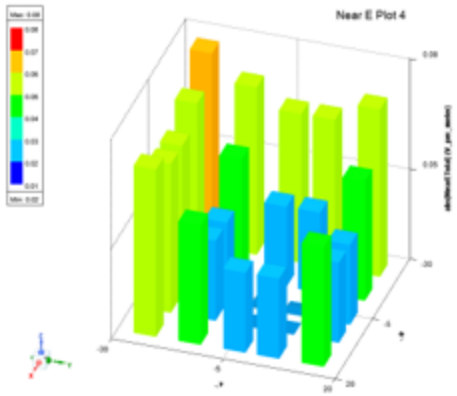
- Transparency



- Filled

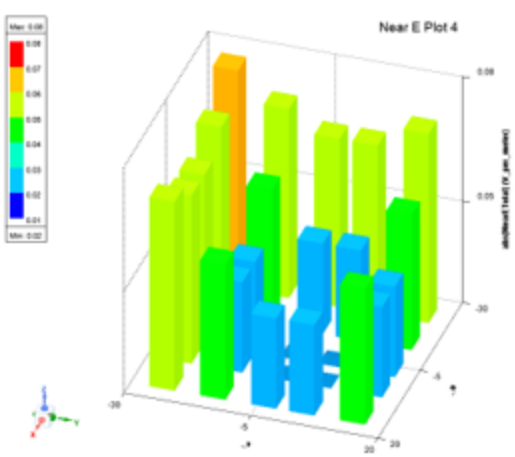


Filled [ ]

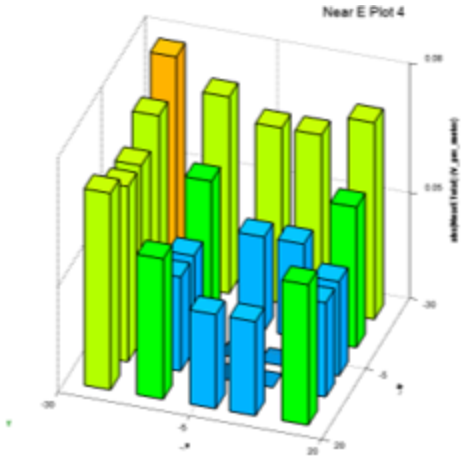


Filled [√]

- Show Outline



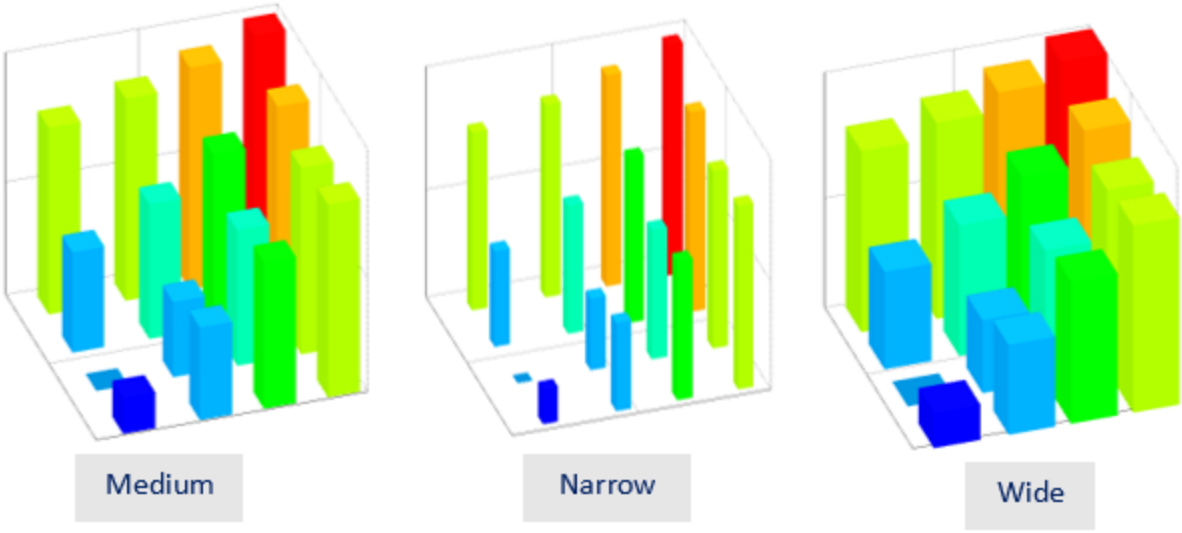
Show Outline []



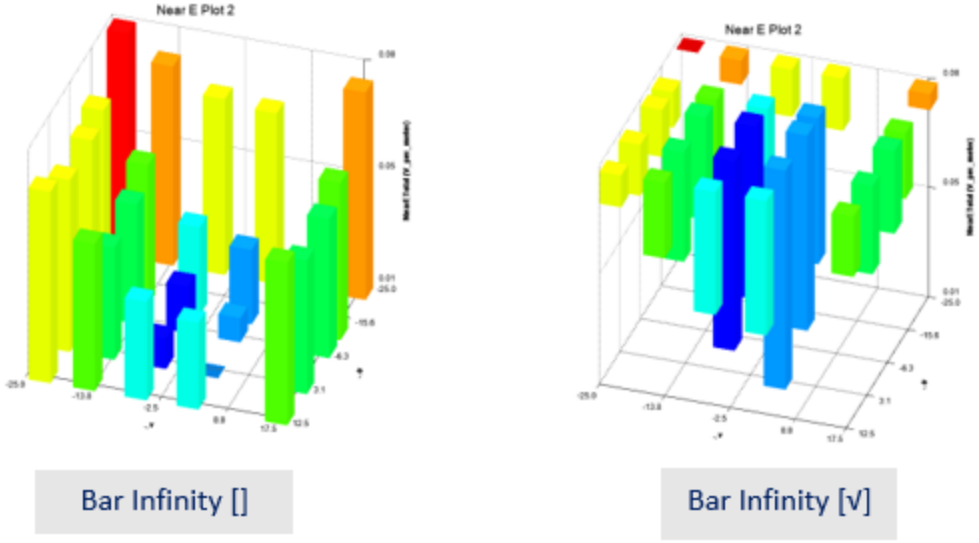
Show Outline [√]

- Outline Color (customizable, see plot above)
- Outline Width (customizable, see plot above)

- Bar Width, controls the thickness of the bar. Medium (default), Wide and Narrow are the width that equals to ½, ¾ and ¼ of the min spacing between adjacent points respectively.

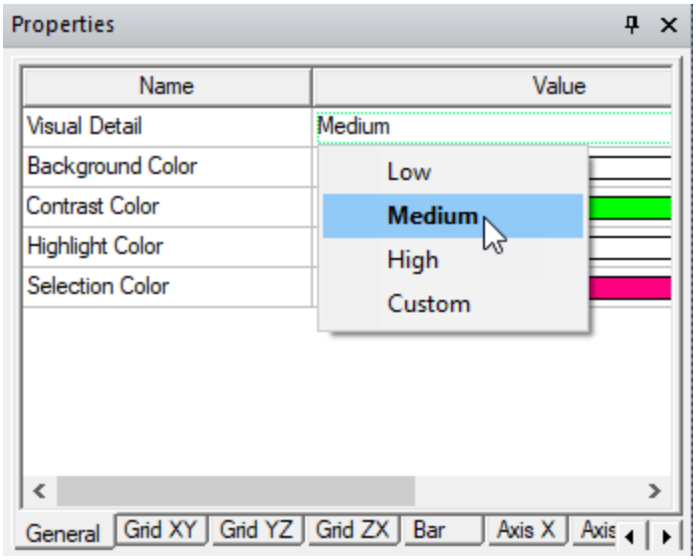


- Bar Infinity represents the third coordinate (z) using the bottom of the bar.



Controlling Visual Detail in a 3D Rectangular Bar Plot

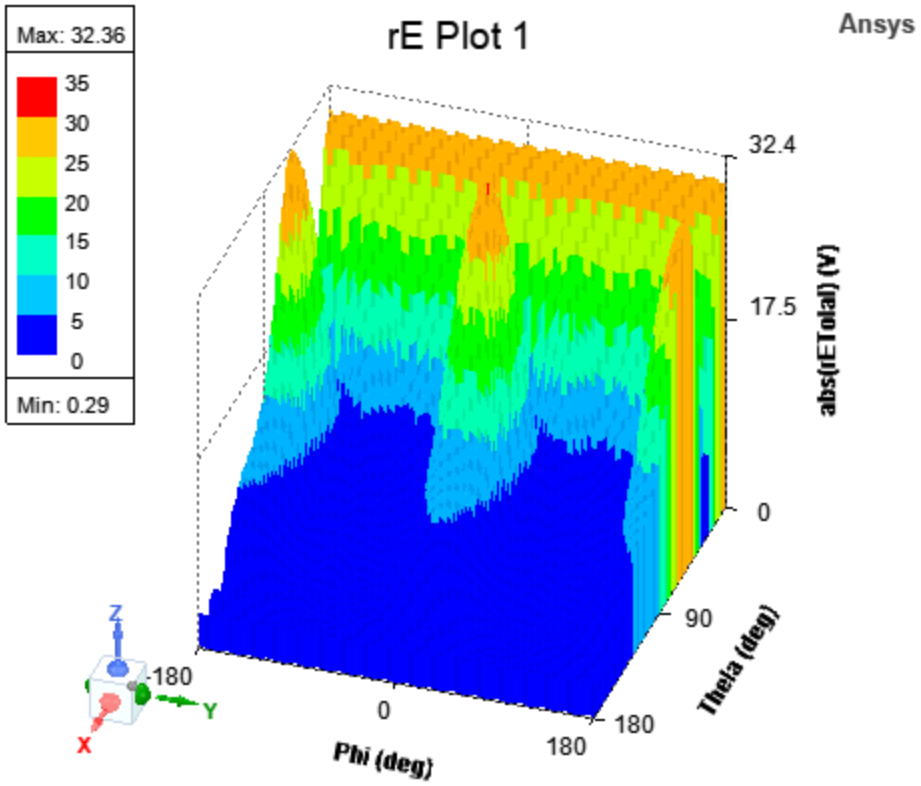
If a particular plot seems busy with information, you can edit plot properties, such as Axis and Grid Attributes for discrete levels of visual detail to improve readability. Double-click anywhere on a plot to display the Properties dialog box. The Visual Detail property on the General tab also provides control suited to different screen and plot sizes.



The Visual Detail menu has four options: Low, Medium (the default), High, and Custom. If you select any Visual Detail, the 3D plot is rendered according to the selected Visual Detail level and the properties reflect the values chosen for the selected visual detail level. From this predefined visual detail level, if you modify any properties, Visual Detail is automatically set to Custom (or to another predefined visual detail level if the edits happen to match the settings for that level).

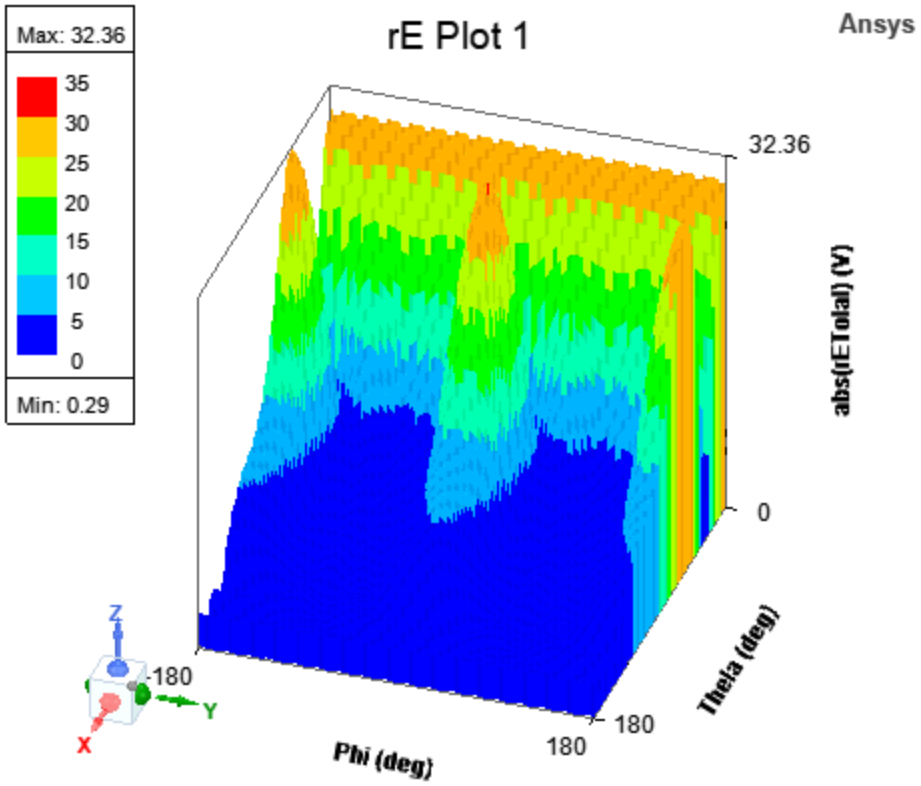
You can also manually set Visual Detail to Custom. In such a case, Custom will inherit property values corresponding to the previous level. This ensures that you can customize settings starting from a baseline provided by the preconfigured Low, Medium or High Visual Detail levels.

3D Rectangular Plot with Medium Visual Detail



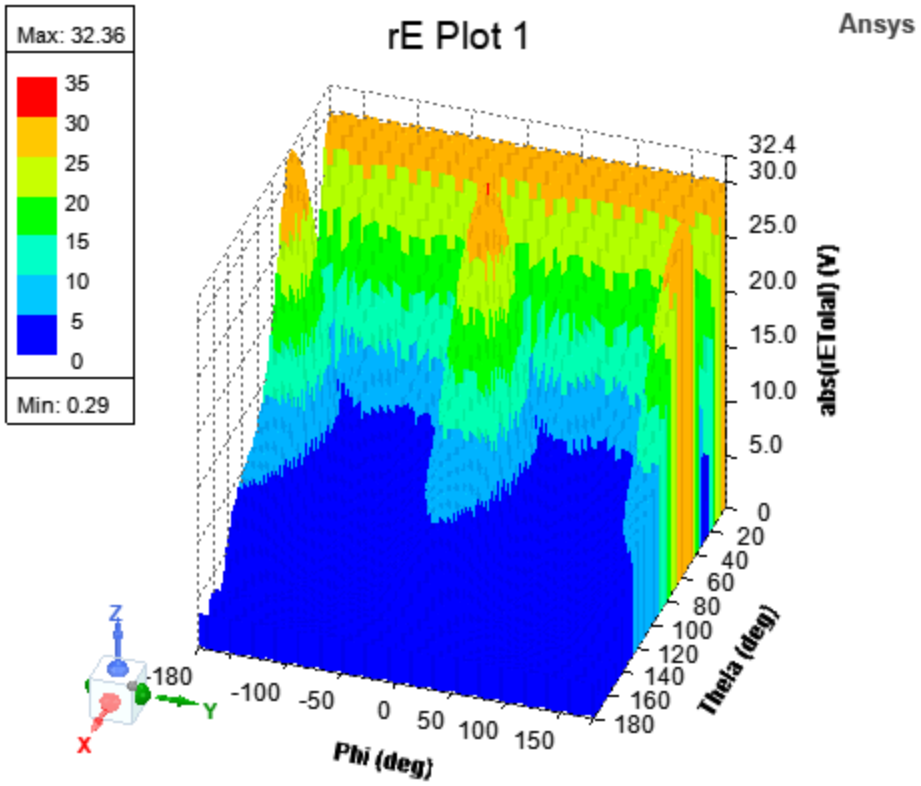
On creation, a 3D Rectangular Plot has Visual Detail set to Medium and looks and feels as shown above. Specifically, under Medium Visual Detail level, a 3D Rectangular Plot has 3 ticks per axis (X, Y, Z axis) which will show min, max and middle value. This setting also shows axes labels.

3D Rectangular Bar Plot with Low Visual Detail



With Visual Detail set to Low, a 3D Rectangular Plot shows axes with 2 ticks corresponding to min and max values. It also shows axes labels and grid borders.

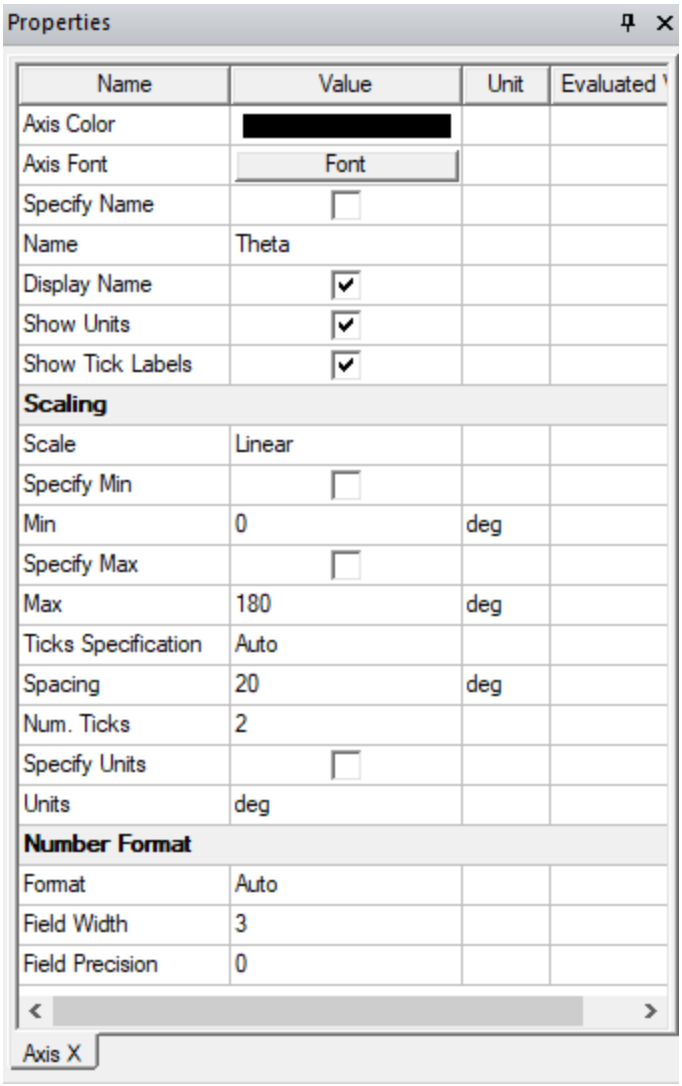
3D Rectangular Bar Plot with High Visual Detail



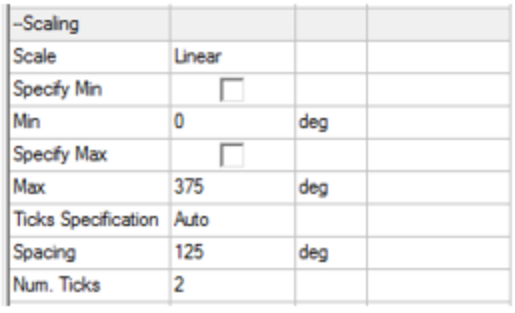
With Visual Detail set to High, a 3D Rectangular Plot shows all Cartesian axes and grids together with all ticks and axes labels.

Axis Properties: Ticks Specification and Num. Ticks

Ticks Specification is available on Axis properties, as shown below:



Ticks Specification is a menu with possible values as Auto, Spacing, and Num. Ticks, with Auto being the default value. If Ticks Specification is Auto, then a spacing value is automatically calculated and used to calculate and display the tick labels. **Spacing** shows the calculated value, and Num. Ticks shows the number of ticks based on this spacing value, as shown below:



You can edit the **Spacing** field when Ticks Specification is set to Spacing; otherwise, it is read only.

You can edit the **Num. Ticks** field when Ticks Specification is Num. Ticks; otherwise, it is read only.

Valid Num. Ticks are between 0 and 100, including 0 and 100. If you enter an invalid value, an error message is shown. If you enter a spacing value that results in number of ticks greater than 100, then an appropriate value is shown.

- If Num. Ticks is 0, then no ticks are shown on the axis.
- If Num. Ticks is 1, then only the max value tick is shown on the axis.
- If Num. Ticks is 2, then only the min and max value ticks are shown on the axis.
- If Num. Ticks is greater than 2, then evenly spaced ticks (including min and max) are shown on the axis.

> **Note:**
>
> With the addition of the Ticks Specification property to Axis properties, the **Specify Spacing** property was removed as an Axis property.
>
> - If an R18.0 or R18.1 project is opened with **Specify Spacing** as Unchecked, Ticks Specification is set to Auto.
> - If an R18.0 or R18.1 project is opened with **Specify Spacing** as Checked, Ticks Specification is set to Spacing.

Related Topics:

Sweeping a Variable

Working with Traces

Adding Characteristics to a Trace

Using the Orientation Gadget

Plotting a Quantity Versus Adaptive Pass

Any quantity that can be plotted in the reporter may be plotted versus the adaptive solution pass, primarily to evaluate the convergence of the quantity versus the mesh.

1. On the **Results** menu (Maxwell menu or right-click on **Results** on the Project tree), click **Create <*type*> Report**, and select **Rectangular plot** from the report type menu.

   The **Report** dialog appears.
2. In the **Context** section, select the **AdaptivePass** context for the Setup of interest. The **Trace** tab **X** value drop down list will contain an option to plot against **Pass**.
3. Under the **Trace** tab, **Y** component section, specify the information to plot along the y-axis:
   a. In the **Category** list, click the type of information (for example, Expression Cache or Output Variable) to plot.

b. In the **Quantity** list, click the value to plot.
c. In the **Function** list, click the mathematical function of the quantity to plot.
d. Value field displays the currently specified Quantity and Function. You can edit this field directly.

Note	Color shows valid expression.

e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.

4. On the **Families** tab, confirm or modify the sweep variables that will be plotted.
5. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog.

   The function of the selected quantity will be plotted against the swept variable values or quantities you specified on an x-y graph. The plot is listed under **Results** in the project tree and the traces are listed under the plot. When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.
6. Optionally, add another trace to the plot by following the procedure above, using **Add Trace rather than New Report**.

Related Topics

Defining Expressions for Non-Transient Solutions
Sweeping a Variable
Working with Traces
Adding Trace Characteristics

Creating Data Tables

A data table is a grid with rows and columns that displays, in numeric form, selected quantities against a swept variable or other quantities.

1. On the **Maxwell3D or Maxwell2D** menu, point to **Results**, and then click **Create <*type*> Report, or right click on the Results icon in the Project tree and click Create <*type*> Report**.
2. In the **display type** menu, click **Data Table**.

   The **Report** dialog box appears.
3. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   - **Solution** field with a drop down selection list. This lists the available setups and sweeps. As a minimum, the LastAdaptive solution and AdaptivePass solution is available to choose.

The AdaptivePass solution context can be selected to allow any value or parameter to be plotted versus the adaptive pass. This function is usually used to evaluate the convergence of the solution.

- *For Transient projects only*:

  Domain field with a pull down selection list containing options for plotting vs time (Sweep Domain) or plotting vs frequency (Spectral Domain).

- *For Transient projects only*:

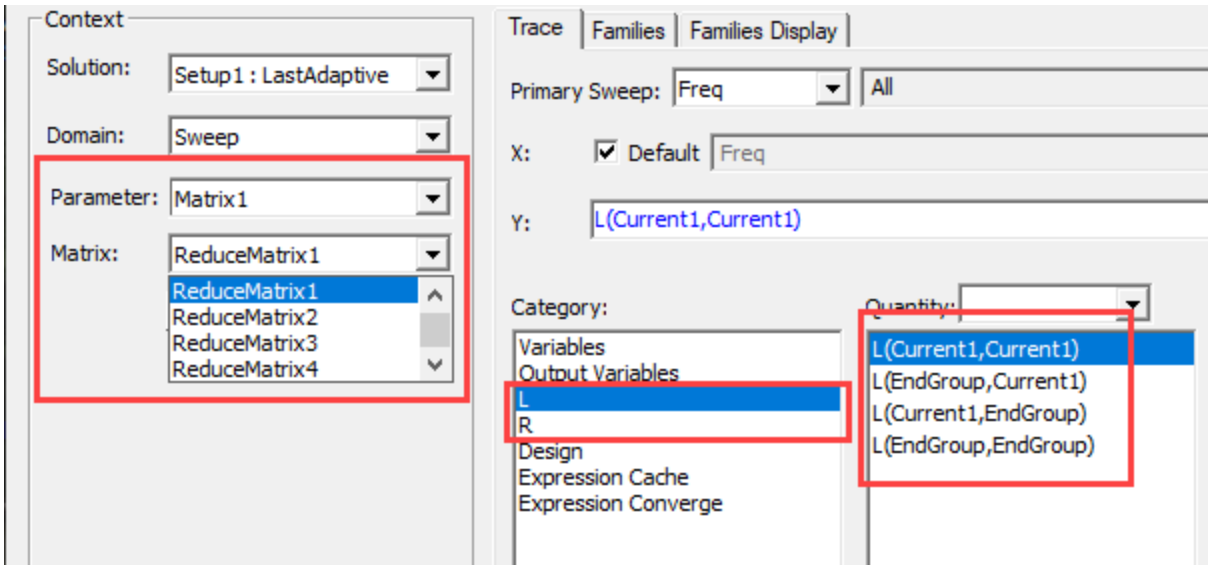
  Domain field with a pull down selection list containing options for **Average and RMS** and **Transient D-Q** for electric machines. Click **Machine Options** to setup the machine specific parameters.

- *For Eddy Current projects only*:

  Domain field with a pull down selection list containing options for plotting vs frequency (Sweep Domain) or plotting vs time (Time Domain).

  Matrix field with a pull down selection list containing options for plotting *matrix* and *reduce matrix* parameters.

Note	In Maxwell Eddy Current designs, the user can create *matrix* parameters, which will cause the solver to produce an impedance *matrix* for the selected excitations. In addition, the user can group (wire) two or more excitations to one excitation in either a series or parallel connection referred to as a *reduce matrix*. The **Matrix** field appears only if a matrix entry is selected in the **Parameter** field. (Refer to *Assigning a Matrix* for information on creating a *matrix* and *Assigning a Reduce Matrix* for information on creating *reduced* matrix parameters for Eddy Current designs.)



- **Parameter** field with a drop down selection list. Whether this field appears, and the parameters listed depend on the Solution type and the ***<type>*** selected.
- *For Fields and Noise Vibration reports*: **Geometry** field with a drop down selection list. This applies the quantity to a specific geometry.

4. Under the **Trace** tab, **Y** component section, select the quantity you are interested in and its associated function:
   a. On the **Category** drop down list, click the type of information to plot.
   b. On the **Quantity** list, click the values to plot. Use CTRL-click to make multiple selections.
   c. In the **Function** list, click the mathematical function to apply to the quantity for the plot.
   d. The **Value** field displays the currently specified Quantity and Function. You can edit this field directly.

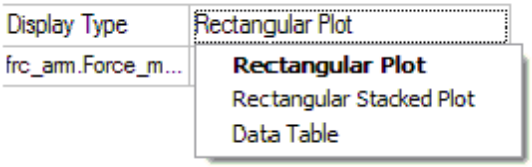
Note	Color shows valid expression.

   e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.
5. On the **Trace** tab, **X (Primary Sweep) line,** select the sweep variable from the drop down list. By default **All** of the chosen sweep's values are used. You can also select the browse [...] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time sweeps), or **Use all values** (the default setting).
6. On the **Families** tab, confirm or modify the sweep variables that will be plotted.
7. Click **New Report**.

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog box. The default name is based on the Report Category you selected, for example, Force Plot *n* or Output Variables Plot *n*. You can edit the plot names in the project tree and the plot header text in the report synchronizes.

   The Y quantity will be listed at each variable value or additional quantity value you specified. The data table is listed under **Results** in the project tree. The plot is listed under **Results** in the project tree and the traces are listed under the plot. When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.
8. Optionally, add another trace to the plot by following the procedure above, using **Add Trace rather than New Report**.

You can also modify the display type of an existing plot from the Properties dialog for that plot. Select the Report icon in the Project tree to display the Properties dialog box. Selecting the Display Type field displays a menu with selections available for that plot.



Once you make a selection, the plot display updates for the current selection.

If you choose to print a data table:

- Selecting print "All" prints the whole table for current data page (if there is more than one data page).
- Selecting print "Pages" prints user-specified pages.
- If the table is bigger than the screen view (that is, it has a scroll bar), printing first scrolls right, prints until no more scrolling occurs, and then scrolls down.
- The Page number appears at the bottom of the page, aligned at center.
- The table layout of each page follows the screen, but without the scroll bar being printed, and no data page bar as on screen.

Related Topics

Sweeping a Variable

Working with Traces

Adding Trace Characteristics

Creating Rectangular Contour Plots

This is an x-y-z graph of results. Any data that you can current plot in 3D is a candidate for a contour plot.

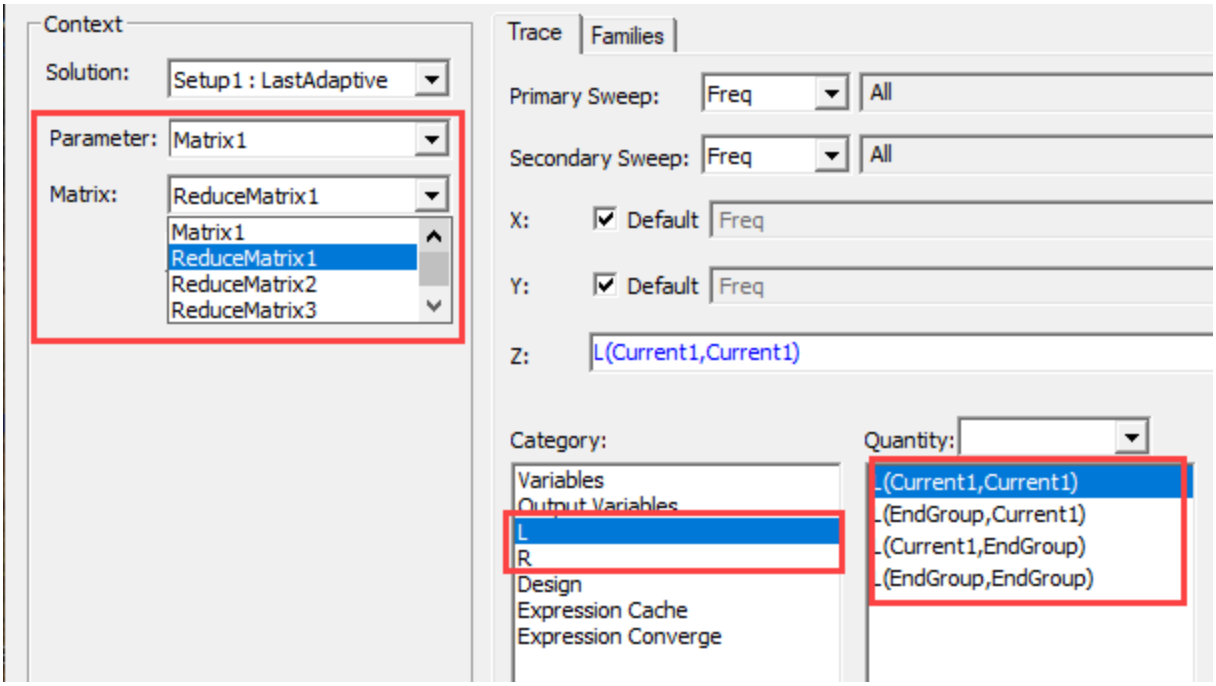
1. On the **Results** menu (Maxwell menu or right-click on **Results** on the Project tree), click **Create <*type*> Report**, and select **Rectangular Contour plot** from the report type menu.

   The **Report** dialog appears.
2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   - **Solution** field with a drop down selection list. This lists the available setups and sweeps. As a minimum, the LastAdaptive solution and AdaptivePass solution is available to choose.

     The AdaptivePass solution context can be selected to allow any value or parameter to be plotted versus the adaptive pass. This function is usually used to evaluate the convergence of the solution.
   - **Parameter** field with a drop down selection list. Whether this field appears, and the parameters listed depend on the Solution type and the ***<type>*** selected.
   - *For Eddy Current projects only*:

     Matrix field with a pull down selection list containing options for plotting *matrix* and *reduce matrix* parameters.

Note	In Maxwell Eddy Current designs, the user can create *matrix* parameters, which will cause the solver to produce an impedance *matrix* for the selected excitations. In addition, the user can group (wire) two or more excitations to one excitation in either a series or parallel connection referred to as a *reduce matrix*. The **Matrix** field appears only if a matrix entry is selected in the **Parameter** field. (Refer to *Assigning a Matrix* for information on creating a *matrix* and *Assigning a Reduce Matrix* for information on creating *reduced* matrix parameters for Eddy Current designs.)



- *For Fields and Noise Vibration reports*: **Geometry** field with a drop down selection list. This applies the quantity to a specific geometry.

3. Under the **Trace** tab **Z** Component area, specify the information to plot as contours:
   a. In the **Category** list, click the type of information to plot.
   b. In the **Quantity** list, click the value to plot.
   c. In the **Function** list, click the mathematical function of the quantity to plot.
   d. The **Value** field displays the currently specified Quantity and Function. You can edit this field directly.

Note	Color shows valid expression.

   e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.
4. On the **Trace** tab **Y** (Secondary sweep) lines, specify the information to plot along the y-axis in one of the following ways:
   - Select the sweep variable to use from the Secondary Sweep drop down list.
   - If sweeps are available, you can select the browse button to display a dialog that lets you select particular values. The quantity will be plotted against the primary sweep variable listed.
5. On the **Trace** tab **X** (Primary sweep) lines, specify the information to plot along the x-axis in one of the following ways:
   - Select the sweep variable to use from the Primary Sweep drop down list.
   - If sweeps are available, you can also select the browse [...] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time

sweeps), or **Use all values** (the default setting). The quantity will be plotted against the primary sweep variable listed.

6. On the **Families** tab, confirm or modify the sweep variables that will be plotted.
7. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog box. The default name is based on the Report Category you selected, for example, Force Plot *n* or Output Variables Plot *n*). You can edit the plot names in the project tree and the plot header text in the report synchronizes.

   The function of the selected quantity or quantities will be plotted against the values you specified on an x-y-z graph. The plot is listed under **Results** in the project tree.When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.
8. Optionally, add another trace to the plot by following the procedure above, using **Add Trace rather than New Report**.

Related Topics

Sweeping a Variable
Working with Traces
Adding Trace Characteristics

Delta Markers in 2D Reports

To view the difference between any two marker points in a report:

1. Set the first marker by left-clicking and holding the mouse button.
2. Move the mouse; without releasing the left button, to another position, and then release the left button to create the second marker.

In the marker text window, the difference between the two markers is displayed instead of the X, Y value of the each marker.

Related Topics

Setting Report2D options
Working with Traces
Editing the Display Properties of Traces
Adding Data Markers to Traces

Working with Traces

A trace in a 2D or 3D report defines one or more curves on a graph. A trace in a data table defines part of the displayed matrix of text values.

The values used for a plot's axes, which may be X, Y, Z, phi, theta, or R depending on the display type, can be variables in the design, such as frequency, or functions and expressions based on the design's solutions. If you have solved one or more variables at several values, you can “sweep” over some or all of those values, resulting in a curve in 2D or 3D space.

A report can include any number of traces and, for 2D Rectangular and Rectangular Contour plots, up to 20 independent y-axes.Traces appear in the Project tree under their report. They can be selected, copied and pasted.

When you move a cursor over a trace in a report, the cursor changes to show that you can make a selection:

- For PC systems, the cursor changes to the color of the selectable trace.
- For Unix systems, the cursor changes to a solid black arrow, rather than the default black outline.

In general, to add a trace to a report:

1. Select a report in the Project window and right-click and select **Modify Report**.
2. In the **Report** dialog specify the Y component information.
   a. Specify the Category of information you want to plot from the drop down menu.
      The Category drop down menu lists the available categories for the Solution type and the current design. Selecting a category changes the Quantity and Function lists to represent what is available for that category.
   b. Specify the Quantity you want to plot by selecting from the Quantity list.
      The selected quantity appears in the Value field, operated on any selected function.
   c. Select the Function to apply to the specified quantity.
   d. The Value field shows the trace being readied for plotting on the Y-axis. This field is editable when the text cursor is present. You can modify the information to be plotted by typing the name of the quantity or sweep variable to plot along an axis directly in the text boxes.

Note	Color shows valid expression.

   e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.
3. In the **Report** dialog specify the X axis information (for example Primary Sweep).
4. Click **Add Trace**.

   A trace is added to the traces list under its report icon in the Project tree. The trace represents the function of the quantity you selected and will be plotted against other quantities or swept variable values. Selecting a Trace in the Project tree displays the Properties window for that Trace.

   Trace icons can be selected, copied, and pasted for their definitions or their data. They can be selected and deleted from the Project tree.

   By default, the Trace name is the definition (the category, quantity and function). The trace will be visible in the report when you click **Add Trace**.

Trace properties can be edited directly in the Properties window or edited in the **Report** dialog box. To change the name of definition of a trace, see Editing Trace Properties. To edit other display properties of a trace, see Editing the Display Properties of Traces.

Related Topics

Removing Traces
Editing Trace Properties
Editing the Display Properties of Traces
Discarding Report Values Below a Specified Threshold
Adding Trace Characteristics
Adding Data Markers to Traces
Setting Report2D options
Copy and Paste of Report and Trace Definitions
Copy and Paste of Report and Trace Data
Delta Markers in 2D Reports

Editing Trace Properties

You can edit trace properties such as the name, the Y Axis association, the component definition, the context, or the variables select the trace in the Project tree.

To edit a trace name:

1. Select the trace in the Project tree.

   This displays a docked Properties window for the Trace.
2. Check the Specify Name box.

   This enables editing of either the Name field in the docked properties dialog box, or the Trace label text in the Project tree. Editing this name changes the display in the Legend and in the Project tree, but not the underlying Y-component definition.

Note	To control the display of the Solution Name and Variation Key in the Legend, see Report 2D: Legend Tab.

To edit the Y Axis associated with the trace (2D Rectangular and Rectangular Contour plots):

1. Select the trace in the Project tree.
2. In the docked Properties window for the trace, select the Y Axis to be associated with the trace from the drop down menu. Up to 20 independent Y axes can be added to a plot.

To edit a trace component definition:

1. Select the trace in the Project tree.
2. In the docked Properties window for the trace, select the component field of interest, and select Edit... from the drop down menu.

This displays the an edit Component field window.form which you can edit the category, quantity and function.

3. Click OK to apply the changes and close the dialog.

To edit a trace Context:

1. Select the trace in the Project tree to display the docked properties window.
2. In properties window, click the Solution field or the Domain field. If other selections are possible, they can be selected from the drop down menu.

To edit a variable for a trace:

1. 1. Select the trace in the Project tree to display the docked properties window.
2. 2. Under the -Variables category, on the Families line, click the Edit button to display the Edit families dialog.

   From this dialog box, you can select the Sweeps or Variations radio buttons.

   If other nominal values are available you can click the ellipsis button to select from a list.

Related Topics

Removing Traces
Editing the Display Properties of Traces
Discarding Report Values Below a Specified Threshold
Adding Trace Characteristics
Adding Data Markers to Traces
Setting Report2D options
Copy and Paste of Report and Trace Definitions
Copy and Paste of Report and Trace Data
Delta Markers in 2D Reports

Editing the Display Properties of Traces

To edit the display properties of a trace:

1. Select a trace in an open **Report** window.
2. Click once on the trace to view a **Docked Properties** window, or double click to open **Properties** window.

   The display properties window for a trace includes a **General** tab and an **Attributes** tab.

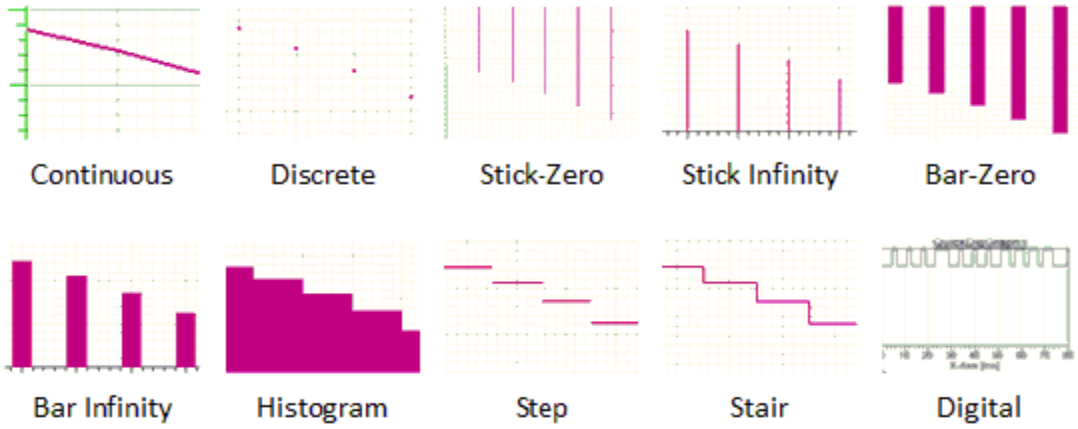
   The **General** tab properties apply to the general appearance of the plot. They include the Background color, Contrast color, Field width, and Whether to use Scientific notation for marker and delta marker displays. (X and Y notation display is set separately, in the Axis property tabs.)

   The Attributes Tab properties apply specifically to the trace. The defaults are set in the Report2D options. They include:

- **Name** – not editable by selecting the trace from the Report. It shows the characteristics of the trace as defined in the Report dialog box.

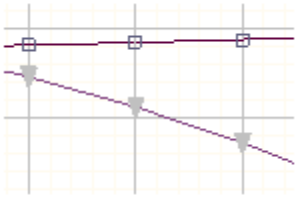
To edit a trace name, see Editing Trace Properties

- **Color** – shows the Trace color. Double click to open a Color dialog box. You can select from Basic colors, or custom colors. You can define up to 16 custom colors by selecting or by editing the Hue, Saturation, Luminescence, and the Red, Green, and Blue values.
- **Line style** – a drop down menu lets you select Solid, Dot, Dash, or Dot-dash.
- **Line width** – a text field lets you edit the numeric value.
- **Trace type** – he drop down menu contains entries for Continuous, Discrete, Bar-Zero, Bar Infinity, Stick Zero, Stick Infinity, Histogram, Step, Stair, and Digital.



Notice the difference between Stair and Digital is that each Stair centers on a data point with transitions halfway between points, and Digital transitions from each data point to the next value.

The next four properties work together to define whether to show a symbol on data points, the symbol frequency, the symbol style, and whether to display the symbol as solid or hollow.



- **Show Symbol** – whether to show a symbol at the data points on the line.
- **Symbol Frequency** – how often to show symbols on the trace, based on the number of data points per symbol used. For example, specify 1 for one symbol per data point. Specify 10 for one symbol for every 10 data points.

- **Symbol Style** – use a drop down menu to select from box, circle, vertical ellipse, horizontal ellipse, vertical up triangle, vertical down triangle, horizontal left triangle, horizontal right triangle
- **Fill Symbol** – use the check box to set the symbol display as a solid or as hollow.
- **Symbol Arrows** – use the check box to use arrows on the curve ends.

Note	So that curves with single points always appear, Box is the default symbol. For Maxwell 12, None cannot be selected.

3. Edit the properties of interest and click **OK** to apply the changes and close the window.

To edit the display properties of a trace in a 3D report:

1. Double click on the trace. This opens a Properties dialog for the plot with a tab named for the trace selected.
2. The editable properties include Point size, Point Style, whether to show points, whether to show line, line width, and line style.

Related Topics

Setting Report2D options
Working with Traces
Editing Trace Properties
Adding Trace Characteristics
Adding Data Markers to Traces
Removing Traces
Discarding Report Values Below a Specified Threshold
Copy and Paste of Report and Trace Definitions

Adding Data Markers to Traces

The Reporter includes **Report 2D>Marker>** menu commands that let you add markers to traces. A marker appears as "m*N*" at the marked point, where *N* increments from 1 as you place additional markers. Each marker can be selected and has editable properties including name, font, background and color. As you place markers, one or more marker legends may be displayed, depending on the **View>Active View Visibility** settings for the legends. The main marker legend appears in the upper left of the plot, and lists the marker names and their X and Y values in a table. You can control the number format for the table values via the Properties window, General tab. Under Marker/Other Number format, you can specify field width, precision, and whether to use scientific notation. This value is independent of the Axis tab number properties. A separate marker legend appears for Delta Markers, as described for the **Delta Marker** command.

When you enter Marker mode, the cursor arrow is accompanied by an "m" while a circle on the selected trace shows the current position for a potential marker.

To end Marker mode, right-click to display the shortcut menu, and select **End Marker Mode**.

The available Marker mode commands and associated icons are the following:

- 
  Add Marker – this command lets you place a marker at an arbitrary point on a selected trace.

- 
  Add Delta Marker – enters delta marker mode, placing a circle on the selected trace. Clicking on the trace sets an initial point and subsequent clicks on arbitrary points on the trace place additional markers until you leave marker mode. These markers have their own legend, which includes the following information for each pair of markers specified.:

Name	Delta(X)	Delta(Y)	Slope(Y)	InvSlope(Y)
d(m2,m3)	0.4700	1.8319	3.8976	0.2566

- **Export Marker Table** – allows you to export the marker table data as a .csv or .tab file.
- **Export Delta Marker Table** – allows you to export the delta marker table data as a .csv or .tab file.

- 
  Add Maximum – places a marker at the Maximum value on the selected trace.

- 
  Add Minimum – places a marker at the Minimum value on the selected trace.
- **X Marker** – this command adds up to 10 movable markers at the origin of the plot with a vertical line rising from the X axis. Each added marker has its own color and editable properties. You can change the X marker value either by dragging the marker, or by editing the X marker property **X value**. To move an X marker, click on the X label and drag it to the desired location. The label at the bottom of the line gives the X coordinate, and a flag on the vertical line identifies the Y coordinate on the trace. A **Lock Drag** trace property lets you lock the drag feature to leave the marker in place. If more than one X marker is present, marker properties include a connecting line between adjacent markers and the delta X value between the markers. The X markers are cleared by the **Clear All** command.
- **Bring X Marker into view** – this command is enabled if an X Marker is not visible in the plot. It allows you to select from a list of existing X Markers to bring into view.
- **Y Marker** – this commands adds up to 10 movable markers at the origin of the plot with a horizontal line extending from the Y axis. Each added marker has its own color and editable properties. You can change the Y marker value either by dragging the marker, or by editing the Y marker property **Y value**. To move a Y marker, click on the Y label and drag it to the desired location. The label at the left of the line gives the Y coordinate, and a flag on the horizontal line identifies the Y coordinate on the trace. A **Lock Drag** property lets you lock the drag feature to leave the marker in place. If more than one Y marker is present, marker properties include a connecting line between adjacent markers and the delta Y value

between the markers. The Y markers are cleared by the **Clear All** command. For more detail on Y Markers and their use, see Y Markers in stacked XY plots.

- **Bring Y Marker into view** – this command is enabled if a Y Marker is not visible in the plot. It allows you to select from a list of existing Y Markers to bring into view.
-  **Next Peak** – moves a selected marker on the next peak on a trace. You must exit marker mode and select a marker to enable this command.
-  **Next Minimum** – moves a selected marker to the next minimum on a selected trace. You must exit marker mode and select a marker to enable this command.
-  **Previous Peak** – moves a selected marker on the previous peak on a selected trace. You must exit marker mode and select a marker to enable this command.
-  **Previous Minimum** – places a marker on the previous minimum on a selected trace. You must exit marker mode and select a marker to enable this command.
- **Go to Start** (Right arrow) – moves a selected trace marker to the first data point. Enabled by leaving marker mode and selecting a marker.
- **Go to Previous** (Left arrow) – moves a selected trace marker to the previous data point.
- **Go to Next** – moves a selected trace marker to the next data point.
- **Go to End** – moves a selected trace marker to the last data point.
- **Next Curve** – selects the next curve in the report, based on the order in the trace legend.
- **Previous Curve** – selects the previous curve in the report, based on the order in the trace legend.
- **Clear All** – clears all markers on a report.

Related Topics

Setting Report2D options

Working with Traces

Adding Trace Characteristics

Removing Traces

Discarding Report Values Below a Specified Threshold

Modifying the Legend in a Report

Editing the Display Properties of Traces

Zooming in or out.

Fitting Objects in the view window.

Showing Objects

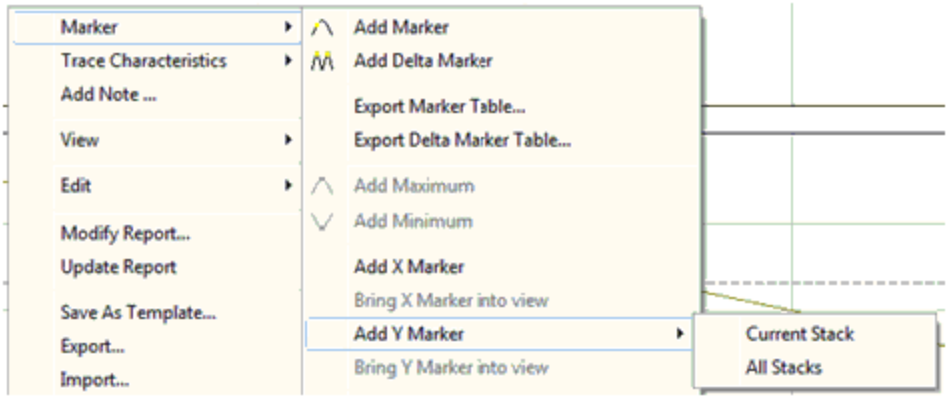
Hiding Objects from View
Delta Markers in 2D Reports
Y Markers in stacked XY plots

Y Markers in stacked XY plots

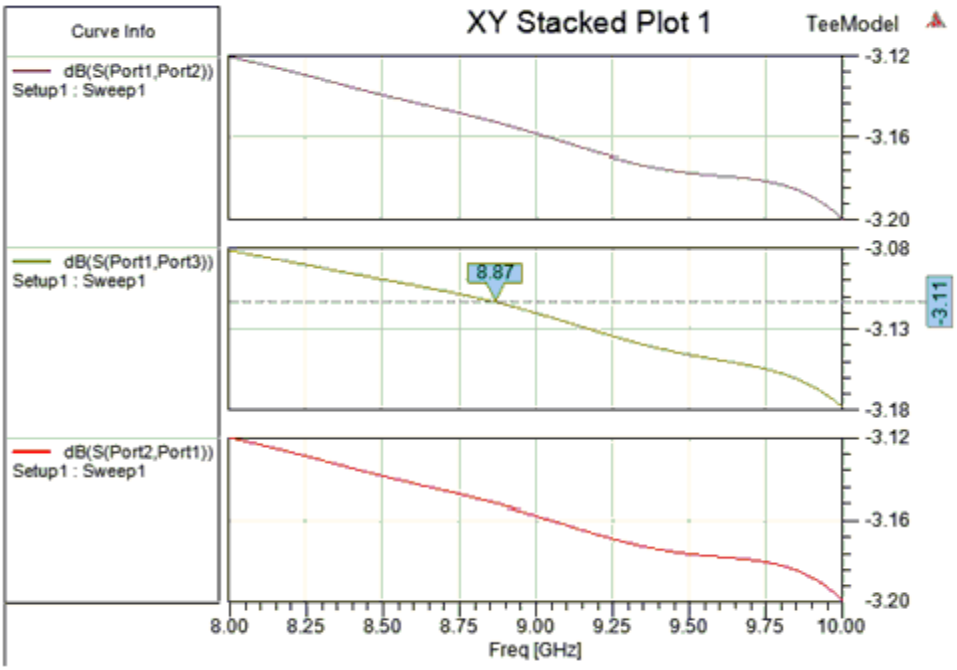
Y Markers allow for easy analysis and comparison of curves at a particular y-coordinate. Y Markers can be used to compare stacked curves.

Creating Y Markers in Stacked Plots

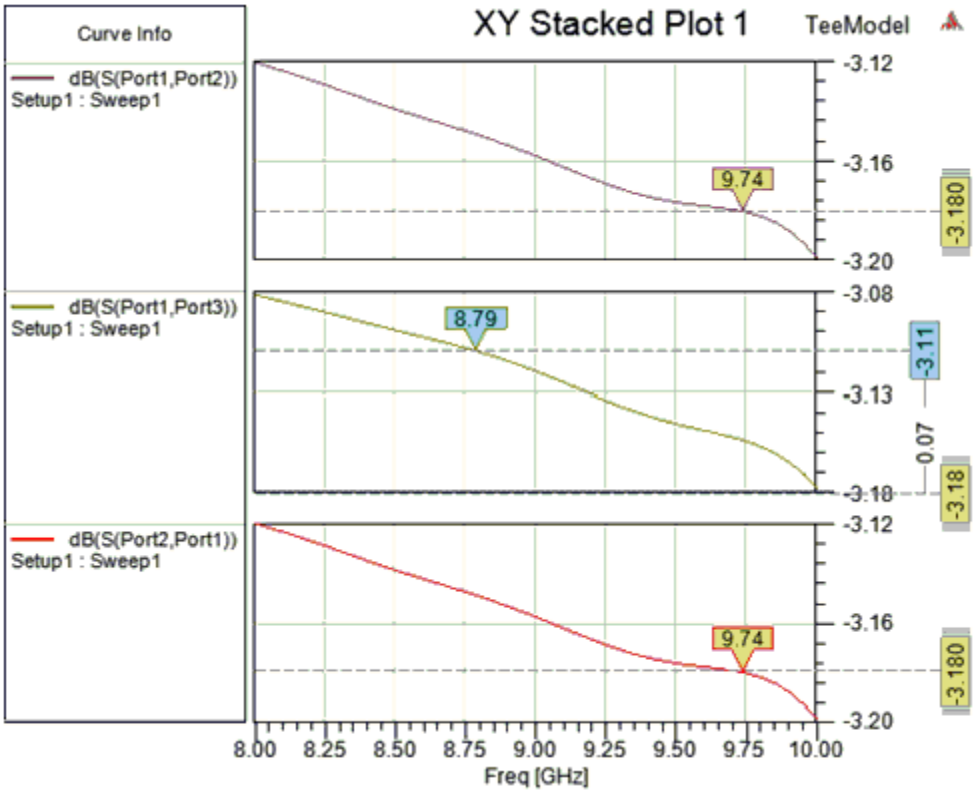
There are two ways to create Y Markers in Stacked Plots. You can create a Y Marker for a particular stack or for all stacks. Right clicking on any stack shows the following context menu:



Add Y Marker > Current Stack creates a Y marker for the stack on which the user performed right mouse button click. The following figure shows that a Y Marker was added to the second stack only:



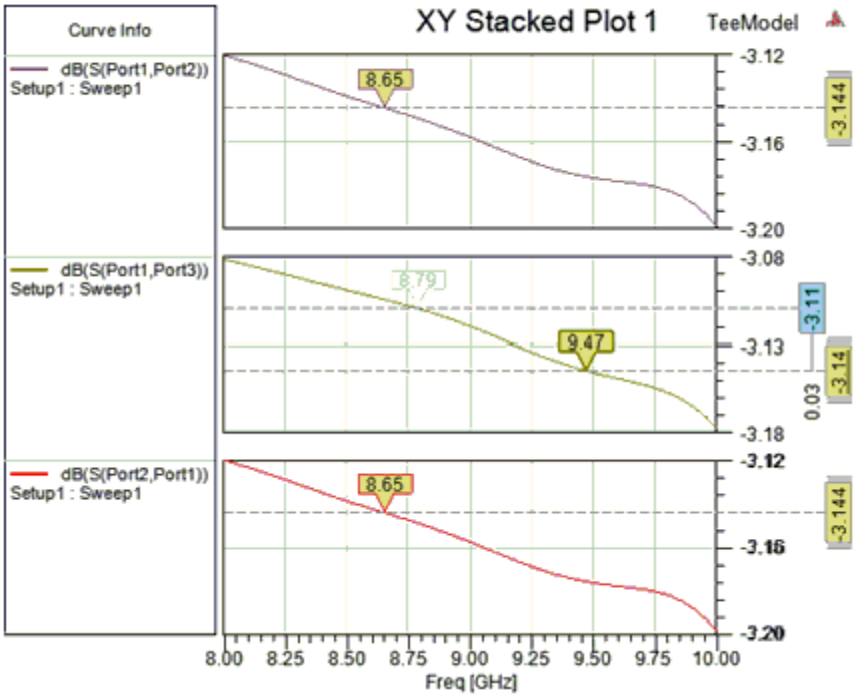
Add Y Marker >All Stacks creates one Y marker in each stack with the same value. Initially this value is the minimum Y value of the Y ranges in all the stacks. This is shown is the figure below:



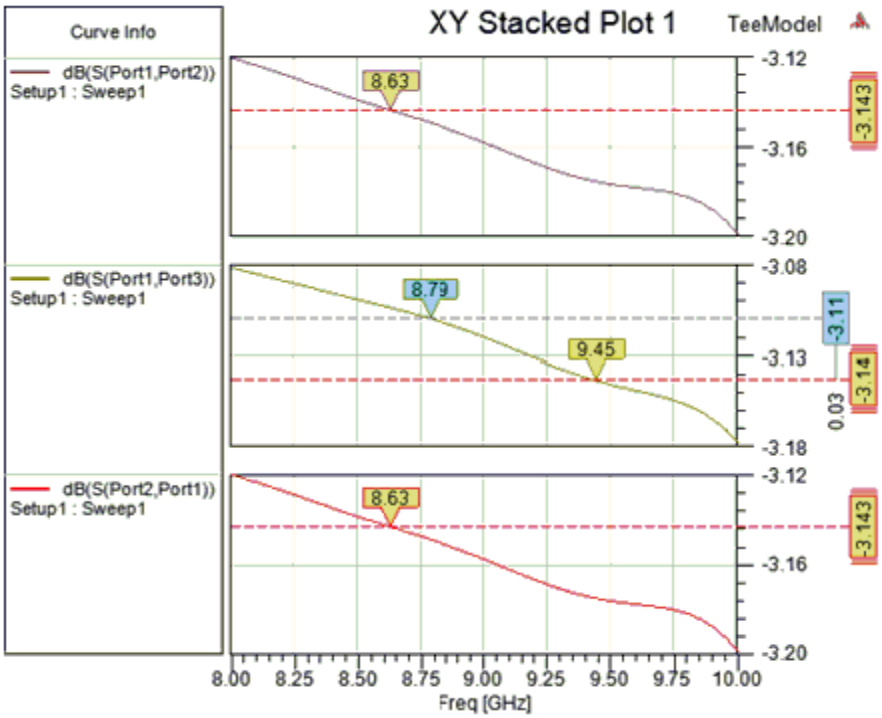
Notice that the Y Marker for All Stacks has a different appearance than the Y Marker for a particular stack: that is, it has double parallel lines above and below the Y Marker textbox.

Synchronized Y Markers

All the "same" Y markers for all stacks are synchronized, that is to say that if one Y marker is dragged or its value is changed, all the "same" Y markers in all the stacks will change their position too. The following figure shows that when a Y marker in the bottom stack was dragged, the Y marker in top stacks moved as well:

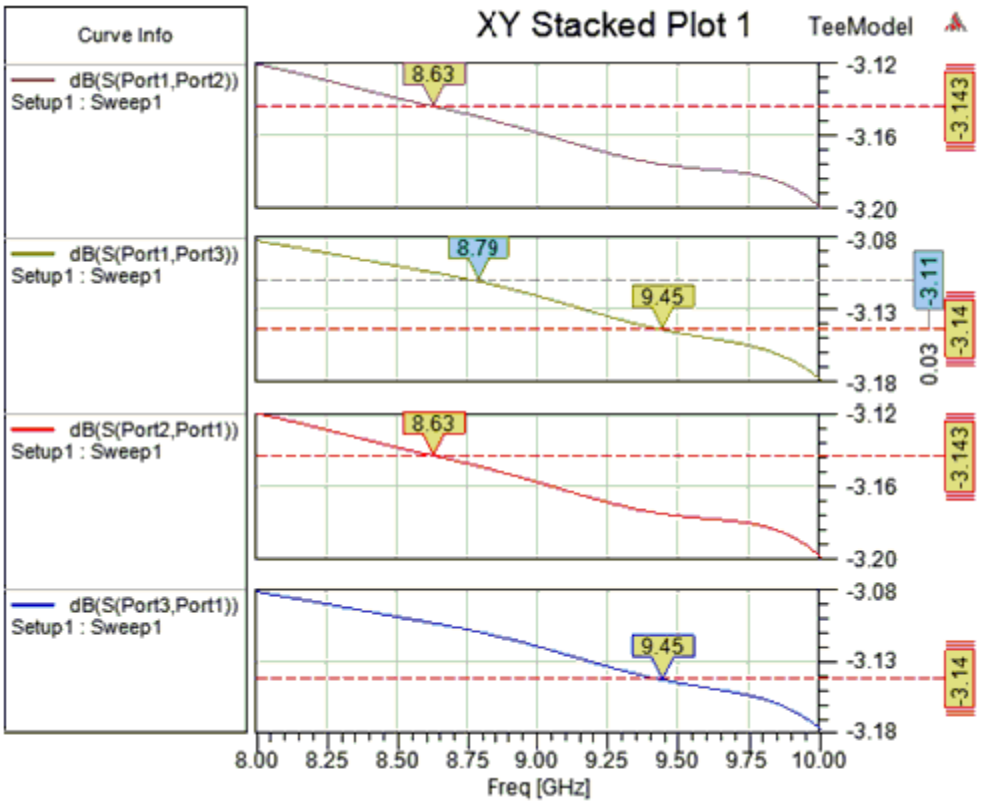


Also if a property of any one Y Marker is changed, all the "same" Y Markers show the change in property as well. For example, the following figure shows that when the line color of a Y Marker in the top stack was changed to a red color, a Y Marker in the bottom stack shows the same line color as well:



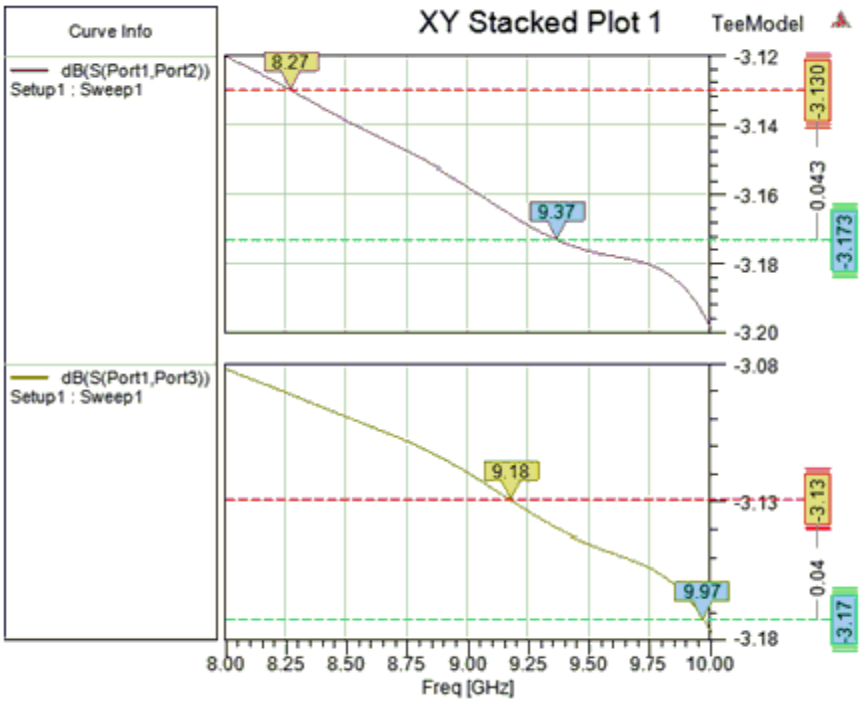
Automatic Y Markers for the new stack

When a new curve is added to the plot, it gets all the Y Markers for all stacks in other stacks, excluding the Y Marker for particular stacks.The following figure shows that when the new curve "dB(S(Port3, Port1)" was added, a Y Marker was added to it with value -3.14 and it has all the same properties as other "same" Y Markers in other stacks:



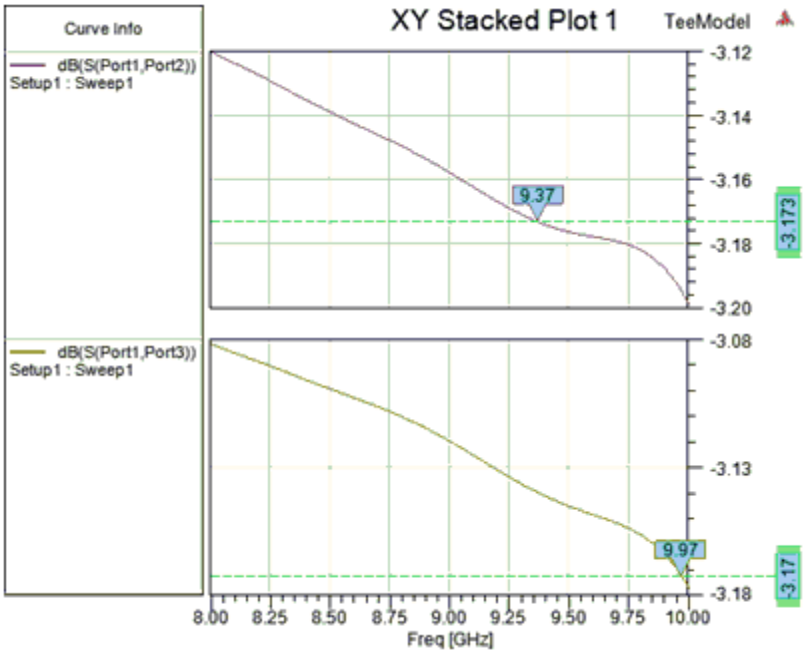
Y Marker Delta Annotations

When two more Y Markers are present in a Stacked Eye Diagram, then delta annotations are shown between a pair of adjacent Y Markers in all the stacks, as shown in the following figure:



Deleting a Y Marker

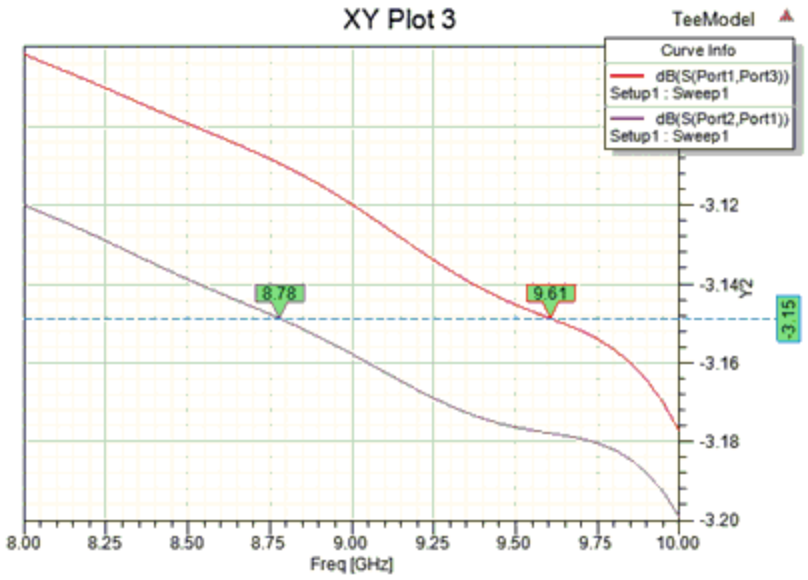
To delete a Y Marker, select a Y Marker in any stack and press the Delete key. This action will also delete all the corresponding Y Markers in all the stacks. For example, when the Y Marker with value -3.13 (red Y Marker) was deleted from the bottom-most stack, all of the corresponding Y Markers were also deleted:



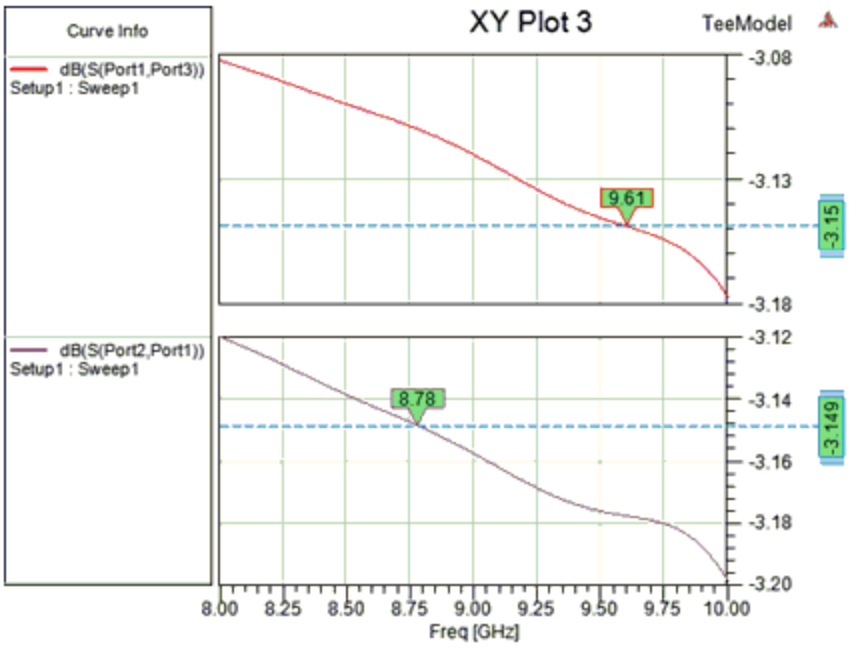
Note that on deleting a stack, Y Markers in other stacks are not affected.

Converting Rectangular XY Plot to Rectangular Stacked XY Plot

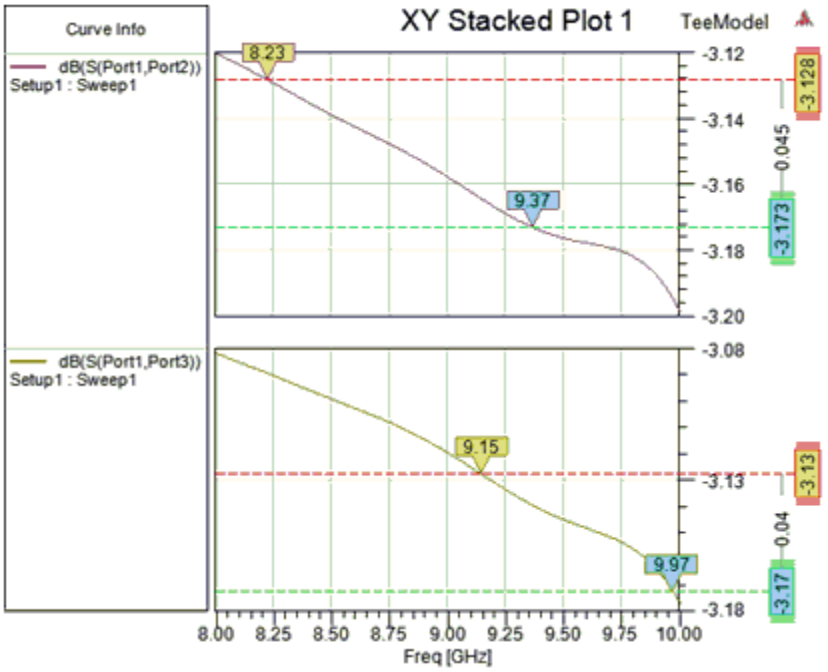
The following figure shows a Rectangular XY Plot with two curves and a Y Marker with value -3.15 (blue Y Marker):



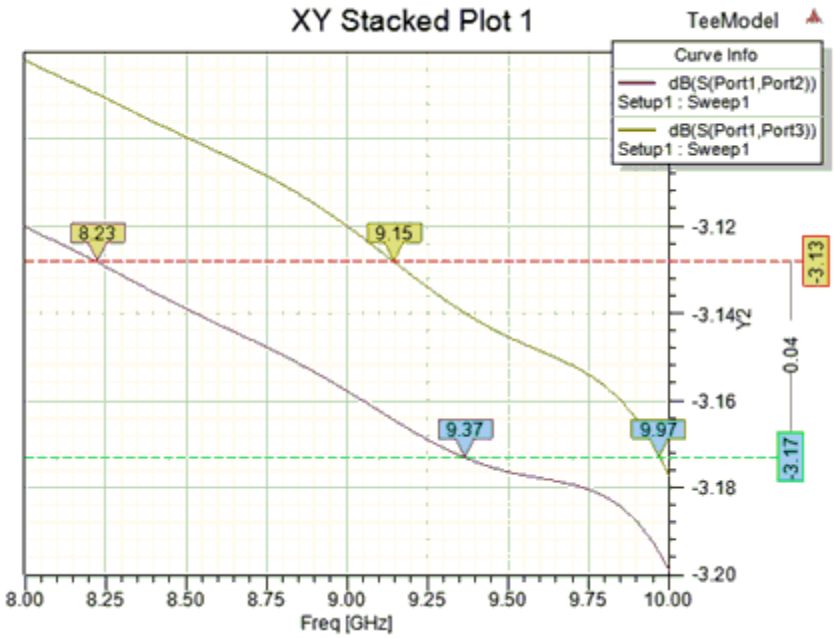
If you change the Display Type property of this plot to Rectangular Stacked Plot, then a Rectangular Stacked XY Plot is created with each curve in its own stack and a Y Marker is shown in each stack with value -3.15 (blue Y Marker):



Similarly when you change a Rectangular Stacked XY Plot to a Rectangular XY Plot, then all the "same" Y Markers in all the stacks are shown as a single Y Marker in the Rectangular XY Plot as shown in following figures:



The Rectangular Stacked XY Plot in the previous figure, when converted to Rectangular XY Plot, looks like the following figure:



Related Topics

Adding Data Markers to Traces
Rectangular Stacked Plot

Discarding Report Values Below a Specified Threshold

To prevent real small numbers from skewing a plot, you can discard small values (below a specifiable threshold).

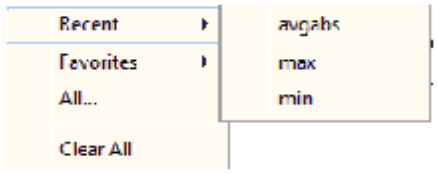
1. Double-click on the X or Y axis of interest on an open plot display.
   This opens the **Properties** window for the Axis
2. Under the **Axis** tab, use the scroll bar to find the **Specify Discard Values** property.
3. Click the check box to enable the property.
4. Enter a value in the **Discard Below** field. Units specified elsewhere in the Axis property are applied to this value. The Discard Below text box is inactive if the Specify Discard Values check box is not enabled.
5. Click **OK** to apply the Discard Values to the report.

Related Topics

Working with Traces
Removing Traces
Editing the Display Properties of Traces
Modifying Background Properties of a Report
Modifying Reports
Adding Trace Characteristics

Adding Characteristics to a Trace

There are several options for adding characteristics to a trace. When you click **Report 2D>Trace Characteristics**, or right-click a selected trace, the short cut menu is displayed. The following example shows the menu with expanded **Recent** selections.



Adding a Recently Used Trace Characteristic

Adding a Trace Characteristic from Favorites
Adding Trace Characteristics to your Favorites
Adding Characteristics using the Add Trace Characteristics Dialog
Removing All Trace Characteristics

Adding a Recently Used Trace Characteristic

If you recently used a characteristic, you can add it to a selected trace by selecting from a list of recently used characteristics. A maximum of 10 is displayed in the menu, and they are sorted alphabetically.

To add a recently used characteristic to a selected trace:

1. Select a trace in a report plot or legend.
2. Click **Report 2D>Trace Characteristics**, or right-click on the selected trace to display the short cut menu.
3. Select **Recent**, and then select the function you want. The specified characteristic is added to the trace.

Related Topics

Adding a Characteristic to a Trace
Adding a Trace Characteristic from Favorites
Adding Trace Characteristics to your Favorites
Adding Characteristics using the Add Trace Characteristics Dialog
Removing All Trace Characteristics

Adding a Trace Characteristic from Favorites

You can add a trace characteristic to a selected trace by selecting from a list of favorites. A maximum of 10 is displayed in the menu, and they are sorted alphabetically.

To add a favorite characteristic to a selected trace:

1. Select a trace in a report plot or legend.
2. Click **Report 2D>Trace Characteristics**, or right-click on the selected trace to display the short cut menu.
3. Select **Favorites**, and then select the function you want. The specified characteristic is added to the trace.

Related Topics

Adding a Characteristic to a Trace
Adding Trace Characteristics to your Favorites
Adding Characteristics using the Add Trace Characteristics Dialog

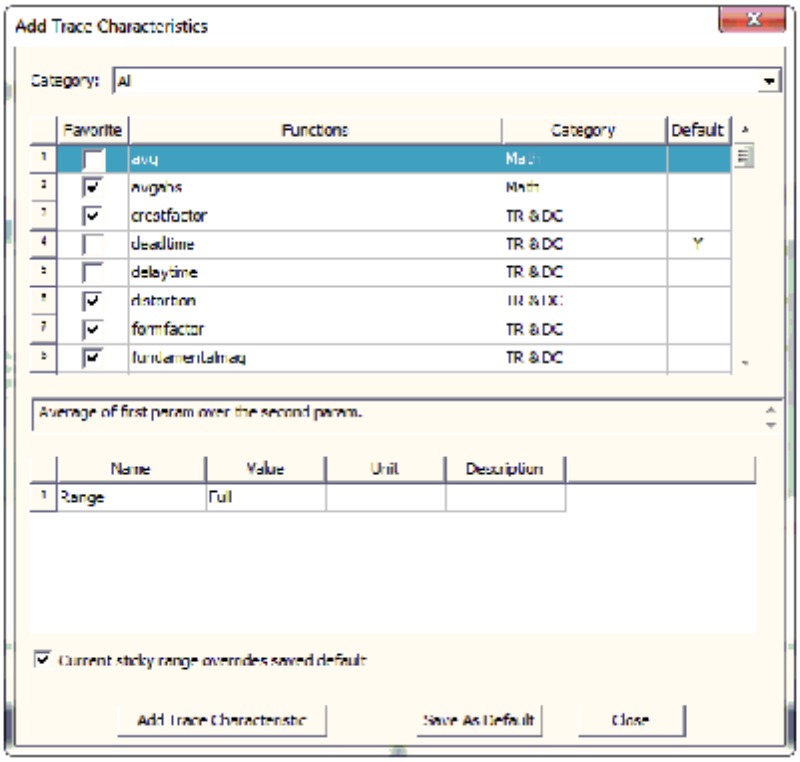
Removing All Trace Characteristics
Adding a Recently Used Trace Characteristic

Adding Trace Characteristics to your Favorites

You can add trace characteristics to your list of favorites.

To add characteristics to your list of favorites:

1. Select a trace in a report plot or legend.
2. Click **Report 2D>Trace Characteristics**, or right-click on the selected trace to display the short cut menu.
3. Select **All**. This displays the **Add Trace Characteristics** dialog box.



4. Click the **Favorite** check box in front of any function you want to add to Favorites. You can define as many favorites as you need, but no more than 10 are displayed in the menu, and they are displayed in alphabetical order.
5. Click **Close**. You can view the current favorites by selecting Favorites in the Category drop-down list

Note	You can remove favorites by clearing the **Favorite** check box for one or more functions, and clicking **Close**.

Related Topics

Adding a Characteristic to a Trace

Adding Characteristics using the Add Trace Characteristics Dialog

Removing All Trace Characteristics

Adding a Recently Used Trace Characteristic

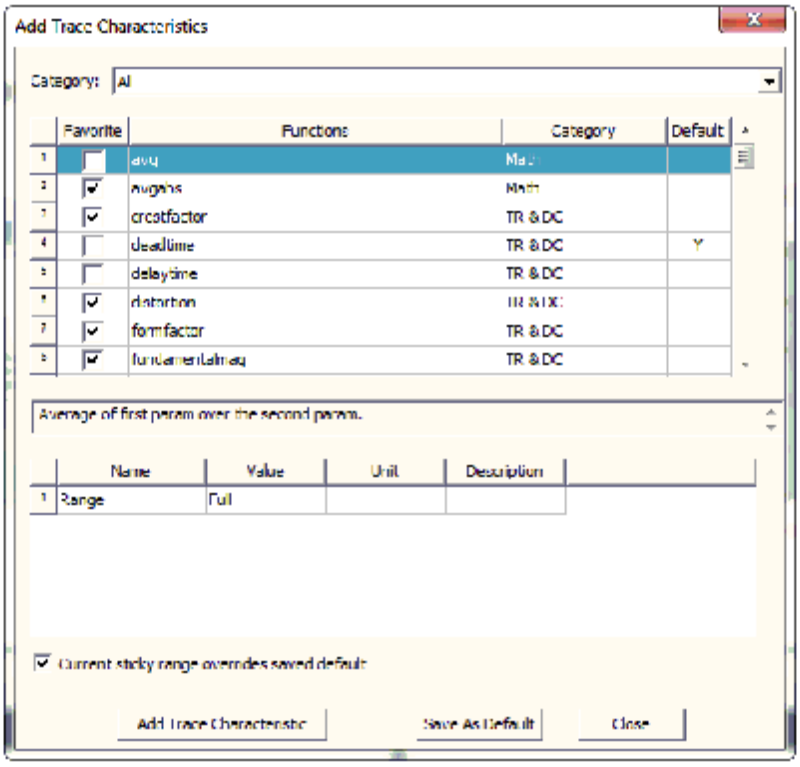
Adding a Trace Characteristic from Favorites

Adding Characteristics using the Add Trace Characteristics Dialog

You can add characteristics to a selected trace by selecting from the **Add Trace Characteristics** dialog box.

To add additional characteristics to a selected trace:

1. Select a trace in a report plot or legend.
2. Click **Report 2D>Trace Characteristics**, or right-click on the selected trace to display the short cut menu.
3. Select **All**. This displays the **Add Trace Characteristics** dialog box.

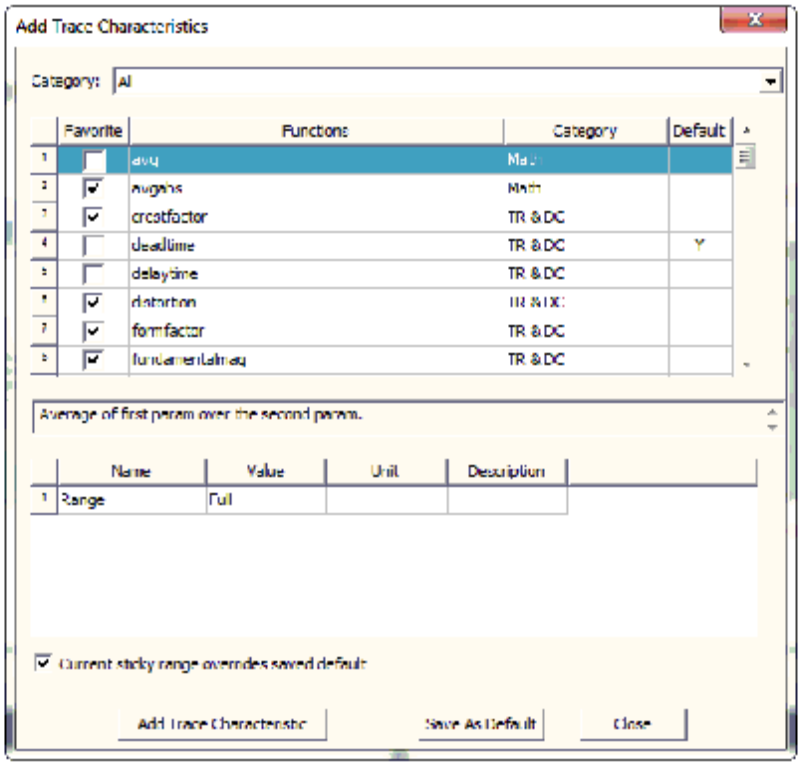


4. Select the **Category**. The available categories depend on the plot, and the selecting of a category displays its associated functions.

Category	Functions
Recent	Displays the most recent functions used, sorted by the time they were added.
Favorites	Displays all favorites. The defaults are avg, max, min, and pk2pk.
All	Displays all available functions.
Math	avg, avgabs, integ, integabs, max, mean, min, pk2pk, pkavg, ripple, rms, rmsAC, stddev, sum, variance, XatYMax, XatYMin, XatYVal, XWidthAtYVal, YatXMax, YatXMin, YatXVal
PulseWidth	pulsefall9010, pulsefront1090, pulsefront3090, pulsemax, pulsemaxtime, pulsemin, pulsemintime, pulsetail50, pulsewidth5050, pw_minus, pw_minus_avg, pw_minus_max, pw_minus_min, pw_minus_rms, pw_plus, pw_plus_avg, pw_plus_max, pw_plus_min, pw_plus_rms
Overshoot/ Undershoot	overshoot, overshootAbs, undershoot, undershootAbs
TR & DC	crestfactor, deadtime, delaytime, distortion, formfactor, fundamentalmag, risetime, settlingtime
Error	iae, ise, itae, itse
Period	per, pmax, pmin, prms
AC	gainmargin, phasemargin, gaincrossover, phasecrossover, lowercutoff, uppercutoff, bandwidth, peakgain, peakgainfreq
Radiation	lSidelobeY, rSidelobeY, lSidelobeX, rSidelobeX, xdb10Beamwidth, xdb20Beamwidth
Eye Measurements	EyeLevelZero, EyeLevelOne, EyeAmplitude, EyeHeight, EyeSignalToNoise, EyeOpeningFactor, EyeWidth, EyeJitterP2P, EyeJitterRMS, EyeRiseTime, EyeFallTime, MinEyeWidth, MinEyeHeight
TDR	Shunt_C_in_pF, Series_L_in_nH

For a selected function, the **Add Trace Characteristics** dialog displays the function's purpose in a text field.

5. Some categories and functions call for you to specify one or two additional values in a table. You can save these values using the **Save as Default** button. The Default column shows a Y if there is a saved default value for the function.



6. Select the **Current sticky range overrides saved default** check box if you do not want the range value in the table to be changed when the function selection is changed: the current range value becomes the “sticky range.” If the check box is not checked, the range value is updated from the saved default values and becomes a new sticky range.
7. Click the **AddTrace Characteristic** button to add the specified characteristics to the trace.
8. Click **Close**.

Related Topics

Adding a Characteristic to a Trace

Adding Trace Characteristics to your Favorites

Removing All Trace Characteristics

Adding a Recently Used Trace Characteristic

Adding a Trace Characteristic from Favorites

Removing All Trace Characteristics

1. Select a trace in a report plot or legend.
2. Click **Report 2D>Trace Characteristics**, or right-click on the selected trace to display the short cut menu.
3. Select **Trace Characteristics>Clear All**.

   Trace characteristics are cleared from the selected trace.

Related Topics

Adding a Characteristic to a Trace
Adding Trace Characteristics to your Favorites
Adding Characteristics using the Add Trace Characteristics Dialog
Adding a Recently Used Trace Characteristic
Adding a Trace Characteristic from Favorites

Removing Traces

You can remove traces from the traces list in the following ways:

To *remove one trace* from the report:

- Select the trace you want to remove from the Project tree, and then click **Delete**.

To *remove all traces* from the report:

- Select all the traces and click **Delete**.

Related Topics

Working with Traces
Editing the Display Properties of Traces

Copy and Paste of Report and Trace Definitions

You can copy and paste report and individual trace definitions within a single design or across designs. The report or trace definition will be evaluated within the context of the target design or report.

Note	If the report or trace definition contains properties that do not exist in the target design (for example, a port name) an error will be posted that indicates a solution does not exist for this trace

Note	You must copy and paste trace definitions between the same report types.

To copy a Report Definition:

Right click on the report name in the project tree and select **Copy Definition** from the shortcut menu.

To paste the Report Definition:

Right click on Results in the project tree of the target design and select **Paste**.

A new report is created and it contains the copied definitions.

To copy an individual Trace Definition(s):

Right click on the trace or traces under a report name in the project tree and select **Copy Definition**.

To paste the Trace Definition(s):

Right click on the report in the target design to which you would like to copy the trace or traces and select **Paste**.

A new trace(s) is added to the report and it contains the copied trace definition(s).

Note	If you copy and paste a report or trace definition to a design which contains a definition with the same name, then an incremented number is appended to the pasted report or trace name.

Related Topics

Copying to the Clipboard as Images

Copy and Paste of Report and Trace Data

Copy and Paste of Report and Trace Data

You can copy and paste report and individual trace data within a single design or across designs. The report and trace definitions and all underlying data within the report or trace are copied and pasted to the target design or report.

To copy all data from a report:

Right click on the report name in the project tree and select **Copy Data**, or use the menu bar **Edit>Copy Data**, or right click within a plot to display a shortcut menu with **Copy Data**.

To paste copied report data:

Right click on Results in the project tree of the target design and select **Paste**.

To copy data from an individual trace(s) in a report:

Right click on the trace or traces under a report name in the project tree and select **Copy Data**.

To paste copied trace data:

Right click on the report in the target design to which you would like to copy the trace data and select **Paste**.

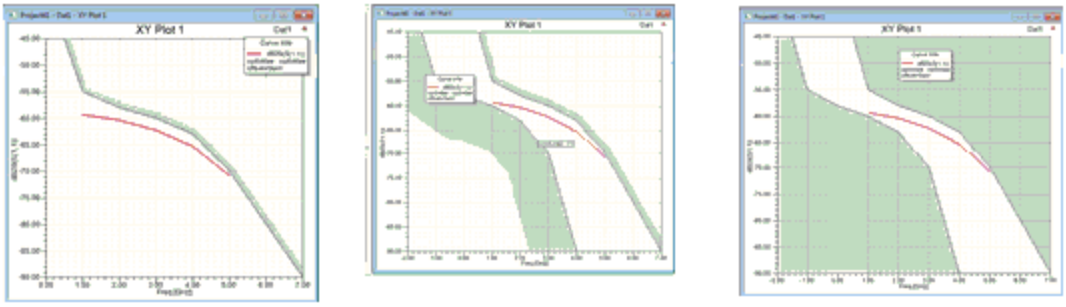
Note	If you copy and paste report or trace data which contains the same name definition as a report or trace in the target design then an incremented number will be appended to the pasted name

Related Topics

Copying to the Clipboard as Images
Copy and Paste of Report and Trace Definitions

Limit Lines in Cartesian Plots

Limit lines are simple graphical representations of constraints on XY plots. These are modeled as a sequence of XY point pairs. For example, you can designate a single limit line to delineate either an upper or lower limit, or two limit lines to delineate upper and lower limits. You can add as many limit lines as you want. You can control the display properties of the line including color and hatch width.



Note	Limit lines are available only on Rectangular (XY) plots - not on XY-like plots such as Bode, Rectangular Stacked, etc.)

You can create limit lines in the following ways:

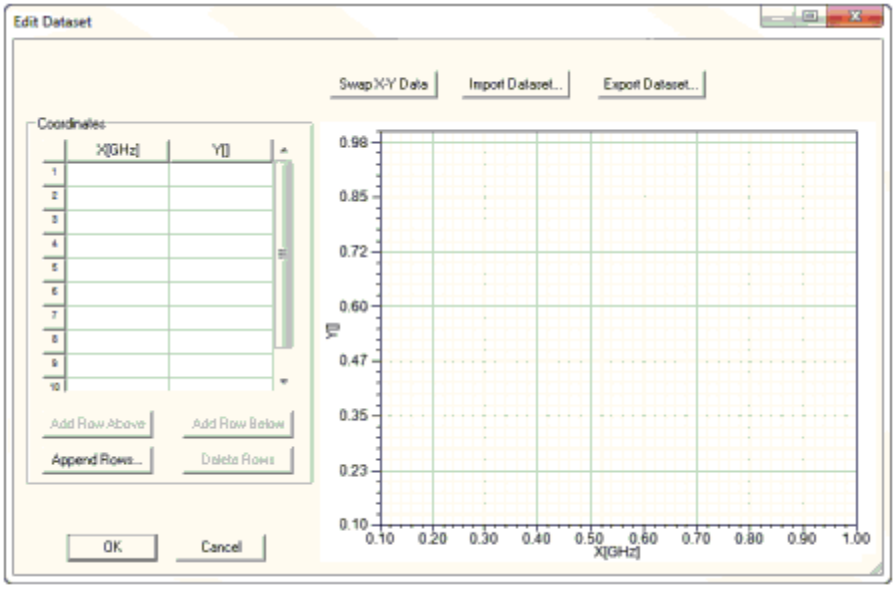
- by specifying points to define the line(s)
- from a selected curve on the plot.

Related Topics

Specifying Points to Create Limit Lines
Creating Limit Lines From a Selected Curve
Editing Limit Lines
Modifying the Background Properties of a Report

Specifying Points to Create Limit Lines

1. Click **Report2D>Add Limit Line>Specify Points** or right-click on an XY plot and select **Add Limit Line>Specify Points** from the context menu to open the **Edit Dataset** dialog.



You can use this dialog to:

- Enter the XY coordinate values directly.
- Import XY values from a tab-delimited **.tab** file.
- Export Dataset to a file.

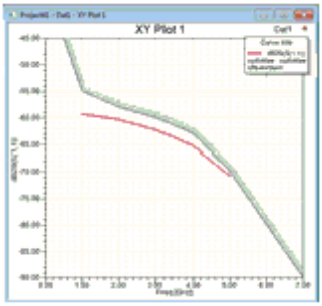
If you require additional data points, you can use the buttons to **Append Rows** to the **Coordinates** table. If you select a row in the Coordinates table, you can then use the buttons to add a row above or a row below the selected rows, or to delete rows.

You can use Shift+click to select multiple adjacent rows, or Ctrl+click to select any combination of rows for deletion.

Note	Each limit line is associated with a particular Y axis (since it has to be scaled the same way as all the curves associated with the axis, follow its log/linear scale, etc.). This Y axis association defaults to the first available Y axis when the limit line is created. However, if the plot contains multiple Y axes, it can be associated with a different Y axis later via its properties tab.

2. When you click **OK**, the limit line you defined is added to the plot. The line divides the plot into regions within the context of its length. By default, the region above the limit line is

hatched.

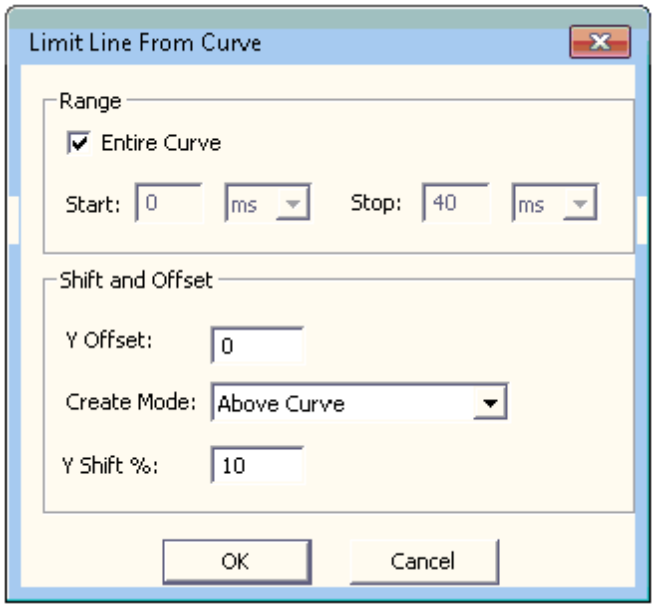


Related Topics

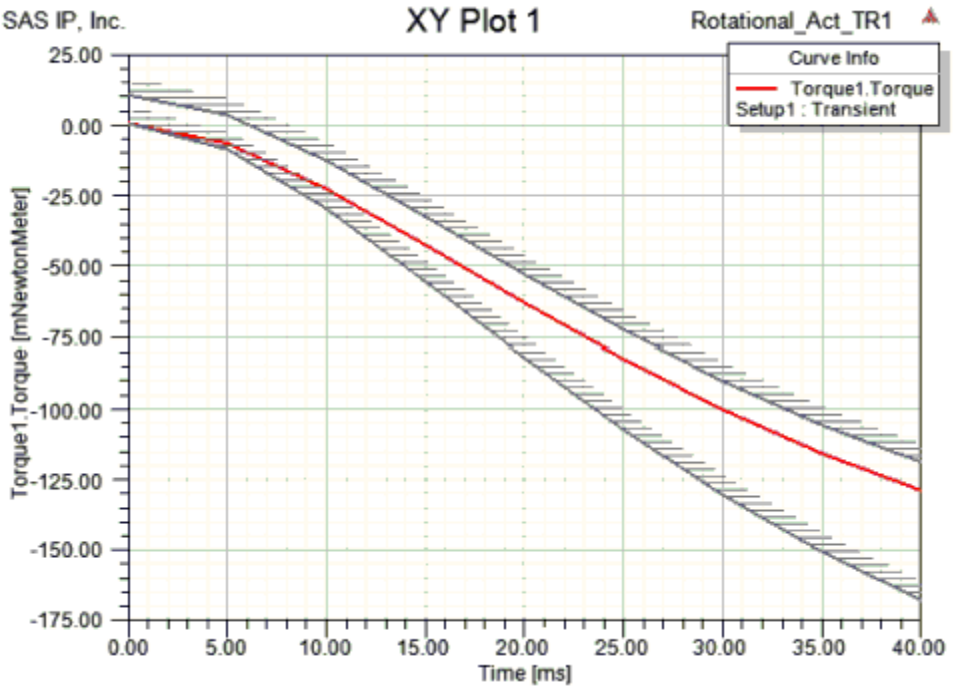
Creating Limit Lines From a Selected Curve
Editing Limit Lines
Modifying the Background Properties of a Report

Creating Limit Lines From a Selected Curve

1. To create limit lines from a selected curve you must first click on the desired plot curve and then select **From Selected Curve** to open the **Limit Line From Curve** dialog.



2. The **Limit Line From Curve** dialog allows you to create limit lines over the **Range** of the **Entire Curve**, or between specified X-axis **Start** and **Stop** points. **Create Mode** controls whether limit lines are **Above Curve**, **Below Curve**, or **Above and Below Curve**. You can also specify a **Y Offset** value, which sets a constant distance between the curve and limit line(s); and a **Y Shift %**, which shifts the limit line(s) by the specified percentage of the Y-axis values. In the following example the upper limit line has a Y Offset value of 0.01 and no Y Shift %, while the lower limit line has a Y Shift % of 30 and no Y Offset,



3. When you click **OK**, the limit lines you defined are added to the plot. The lines divide the plot into regions within the context of its length. By default, the region above a limit line is hatched.

Related Topics

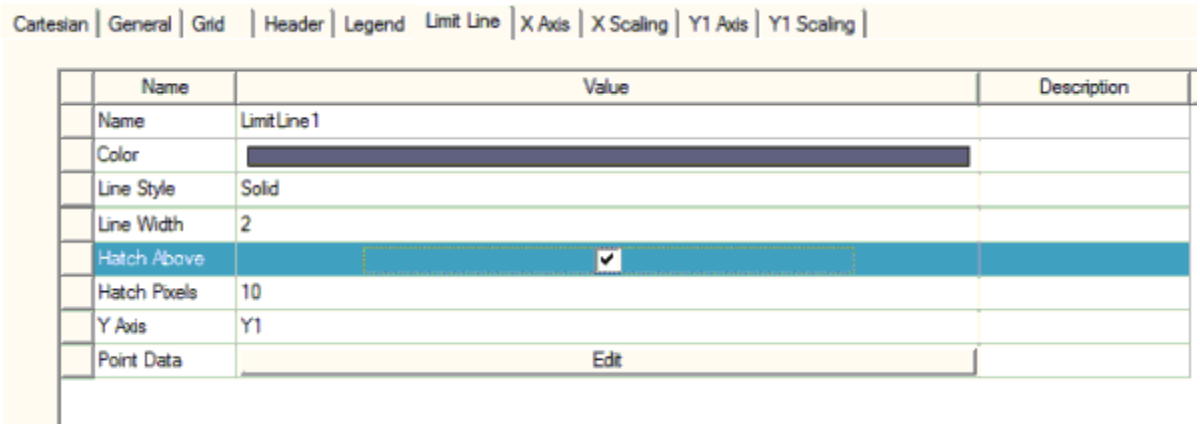
Specifying Points to Create Limit Lines

Editing Limit Lines

Modifying the Background Properties of a Report

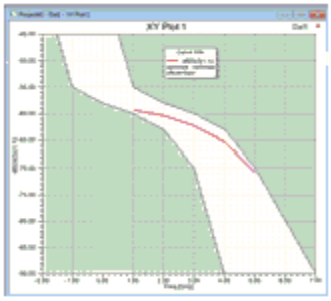
Editing Limit Lines

You can select a limit line in a plot to edit the following properties on the **Limit Line** tab of the plot properties dialog.



- Color – affects both line and hatching
- Style and Width – affects only the line, not the hatching
- Y Axis Y– axis associated with the limit line
- Hatch Pixels – sets the length of the hatch lines
- Point Data – edit the data that defines the limit line
- Hatch Above – sets the direction of the hatch lines (above or below) the limit line

For example, if you add a second limit line, you could designate it as hatch below by unchecking **Hatch Above** to produce a “tunnel” marking the upper and lower constraints.



Related Topics

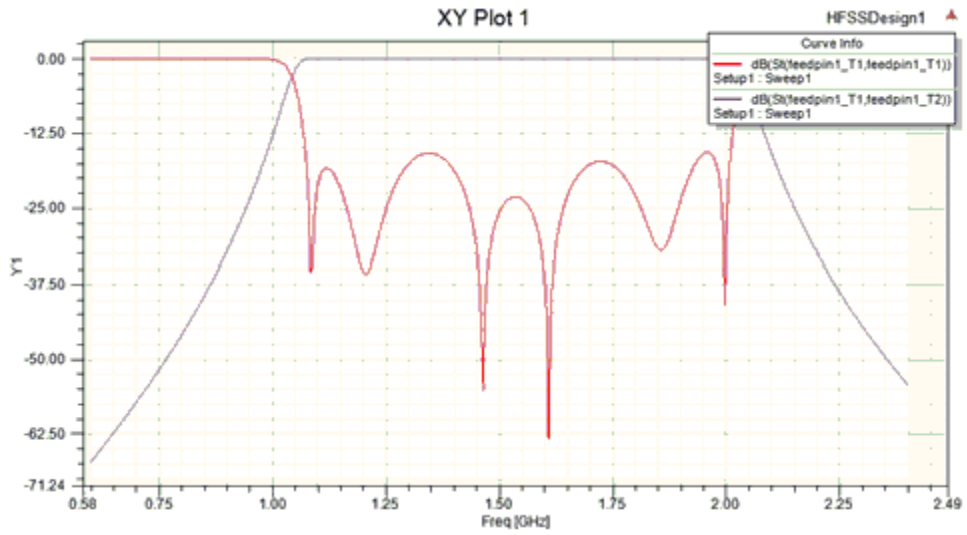
Specifying Points to Create Limit Lines

Creating Limit Lines From a Selected Curve

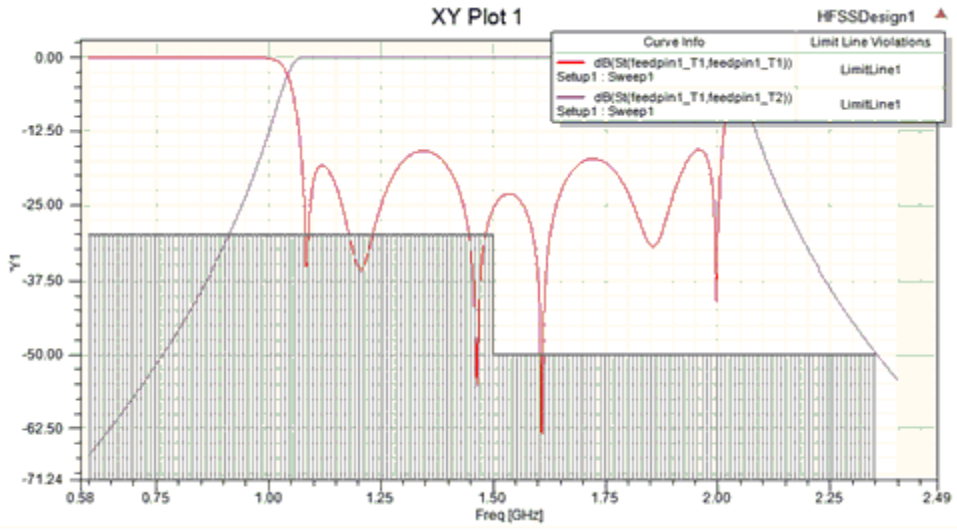
Modifying the Background Properties of a Report

Limit Line Violations

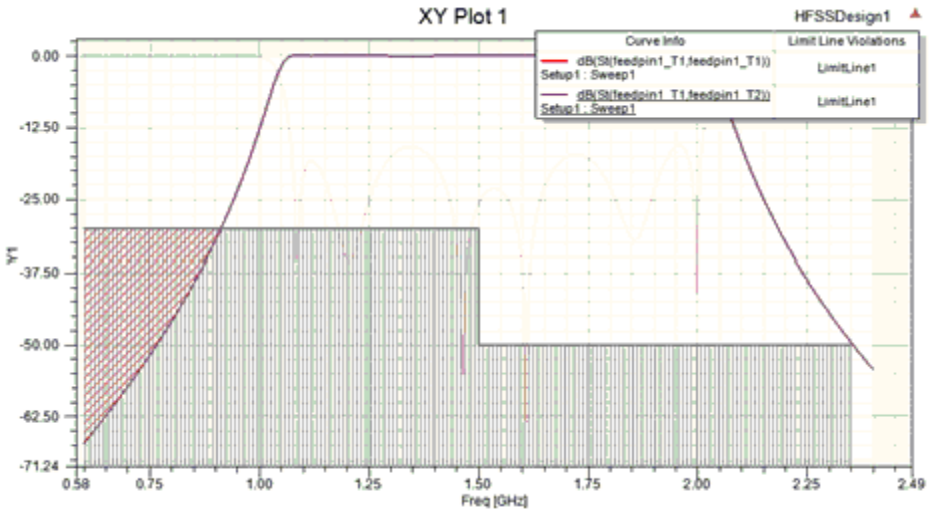
A plot feature can help you detect whether or not a curve violates the limits set by a limit line. Consider the following plot, which shows two curves:



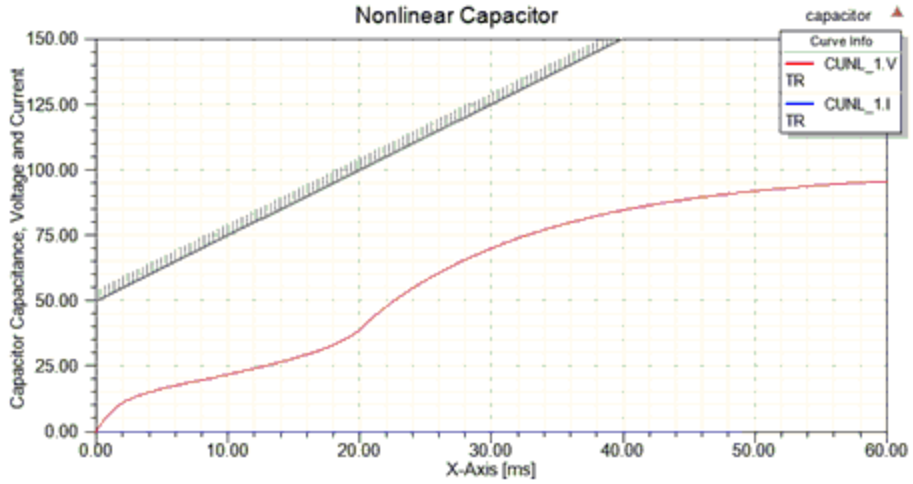
Suppose that the desired response cannot be below -30 dB at frequencies below 1.5 GHz and below -50 dB above 1.5 GHz. If you add a limit line for these requirements in the plot, the plot automatically calculates whether a curve violates this limit line requirement and displays it in the legends window, as shown in following figure:



If a curve is selected, the plot shows the region of the curve that violates the limit line (shaded with slanted red lines), as shown in following figure:



If no curve violates a limit line, the limit line hatching is limited to 10 pixels, as shown in the following figure. This is done to retain focus on the curves instead of the limit lines.



Error Handling

If plot encounters an error while calculating Limit Line Violations, the legend window shows NaN (Limit Line name) in front of the curve in the Limit Line Violations column. This ensure that you do not get misleading information about no-violations when the issue is actually a limitation in our code.

Related Topics

Modifying the Background Properties of a Report

Sweeping a Variable in a Report

In Maxwell, a swept variable is an intrinsic, project, or design variable that typically has more than one value. You can plot any calculated or derived quantity against one or more of the swept variable's values.

To specify the swept variable values to plot a selected quantity against:

1. In the **Report** dialog box, select the variable from the X (Primary Sweep) pull-down menu.
2. To modify the values that will be plotted for a variable:
   a. Click the ellipsis **[...]** button on the **X (Primary Sweep)** line of the **Report** dialog to displays a popup list of the possible values.
   b. Select **Use all values** or **Select values** and click the **Edited** button to display a dialog that lets you specify the sweeps to use.
   All of the selected variable's values will be plotted.

Note	When specifying a Time sweep for a solution that is defined with varying save field, it is recommended that you select **Use all values**.

Sweeping Values Across a Distance

1. If you are plotting a field quantity along a line, define a polyline object in the problem region.
2. In the **Report** dialog box, click the line geometry of interest in the **Geometry** list.
3. Specify the quantities you want to plot along the axes.
4. For the **X (Primary Sweep)**, select the **Distance** variable.
   The values at which the selected quantity or quantities will be plotted are listed to the right. By default, a post-processing polyline object is divided into 100 equally spaced points.

Related Topics

Sweeping a Variable in a Report

Selecting a Function

The value of a quantity being plotted depends upon its mathematical function, which you select from the **Function** list in the **Report** dialog box. The available, valid functions depend on the type of quantity (real or complex) that is being plotted. The function is applied to the quantity which is implicitly defined by all the swept and current variables. "Volume(Stator)" is the volume of the model object named Stator for every swept combination of variables ("height" and so forth).

These functions can also be applied to previously specified Quantities and Functions as **Range Functions** when using the **Set Range Function** dialog.

Some of these functions can operate along an entire curve. These are: min, max, integ, rms, pk2pk, cang_deg and cang_rad.

You can select from the following functions in the **Trace** tab **Function** list or type them directly into the Y or X field, if necessary:

abs	Absolute value of the simulation quantity which results in a number that is always positive.
acos	Arc cosine (the inverse function of a cosine).
acosh	Inverse hyperbolic arc cosine.
ang	Magnitude of an angle.
ang_deg	Angle (phase) of a complex number, cut at +/-180. Returns angular values in degree units, and not suitable for use in Optimetrics, which works in SI values and evaluates ang_deg expressions in radians. See ang_deg_val.
ang_deg_val	Angle (phase) of a complex number in unitless degree values, and suitable for use in Optimetrics, which works in SI values. Returns simple numbers.
ang_rad	Angle in radians.
arg	Argument of a complex number. It is the angle the complex number makes with the positive x axis. Same as **ang_deg**.
asin	Arc sine (the inverse function of sine).
asinh	Inverse hyperbolic sine.
atan	Arc tangent (the inverse function of a tan).
atanh	Inverse hyperbolic tan.
atan2	Two argument function. For non-0 x,y, the function returns the angle between the + x-axis and the given x,y coordinates.
avg	Returns the average of the values of the selected quantity. avg = (Area between the curve and the X-axis) / (X length of the curve)
avgabs	Returns the mean of the absolute value of the selected quantity.
bandwidth	Returns the 3dB bandwidth of the selected simulation quantity. For bandwidth, the calculation is based on 3dB below the maximum peak.
cang_deg	Cumulative angle (phase) of the first parameter (a complex number) in degrees, along the second parameter (typically sweep variable). Returns a double precision value cut at +/-180. Returns angular values in degree units, and not suitable for use in Optimetrics, which works in SI values and evaluates cang_deg expressions in radians. See cang_deg_val.
cang_deg_val	Cumulative angle (phase) of the first parameter of the selected simulation quantity in unitless degree values and suitable for use in Optimetrics, which works in SI values. . Returns simple numbers.
cang_rad	Cumulative angle of the first parameter in radians along a second parameter (typically a sweep variable). Returns a double precision value.
cmplx(*re*, *im*)	A complex number, where *re* is the real part and *im* is the imaginary part.

conjg	Conjugate of the complex number.
cos	Cosine.
cosh	Hyperbolic cosine.
crestfactor	Returns the crest factor (peak/RMS) for the selected quantity.
cum_integ	The cumulative integral function returns a set of values that have the same length as the original set of points (the first element will always be zero). Element *I* of the set returned by cum_integ is the integral of elements 1 through *I* of the original data set.
cum_sum	The cumulative sum function returns a data set that has the same length as the original set of points. Element *I* of the set returned by cum_sum is the sum of elements 1 through *I* of the original data set.
dB(x)	20*log10(|x|) to base 10.
dBc	Decibels relative to the carrier. It is the power ratio of the signal to a carrier signal. Gives the relative signal strength.
dBm(x)	10*log10(|x|) +30.
dBm	(for electric field quanitities) is computed as: 20.0*log10(x) + 60.0
dBu	(for electric field quanitities) is computed as: 20.0*log10(x) + 120.0
dBW(x)	10*log10(|x|).
dB10	10*log(|x|) to base 10.
dB10normalize	10*log [normalize(mag(x))].
dB20	20*log(x) to base 10.
dB20normalize	20*log [normalize(mag(x))].
deadtime	Obtains the latest time when the qtyl is within a tolerance of zero.
delaytime	Obtains the time from zero to 50% of the target point.
degel	Conversion from degrees electrical to seconds with respect to Hz.
deriv	Derivative of a given parameter.
distortion	Returns the total distortion for the selected simulation quantity and an additional argument frequency, which is the frequency in Hz at which to calculate the fundamental RMS of the simulation quantity.
even	Returns 1 if integer part of the number is even; returns 0 otherwise.
exp	Exponential function (the natural anti-logarithm) of the simulation quantity.
formfactor	Returns the form factor (RMS/Mean Absolute Value) for the selected quantity.
fundamentalmag	Returns the RMS value of the fundamental frequency for the selected quantity, and an additional argument, Frequency, which specifies the fundamental frequency.
gaincrossover	Returns the gain crossover frequency (where the gain is 0 dB) of the selected simulation quantity in Hz.
gainmargin	Returns the gain margin in dB at the phase crossover frequency of the

	selected simulation quantity. It also requires a reference simulation quantity to which the measured quantity is compared and the AC magnitude and phase angle of the reference quantity. These are entered as the arguments Reference Channel, Base Source Magnitude, and Base Source Angle.
iae	Returns the integral of the absolute deviation of the selected quantity from a target value that is entered via the additional argument.
if	if(cond_exp,true_exp, false_exp).
im	Imaginary part of the complex number.
int	Truncated integer function.
integ	Integral of the selected quantity. Uses trapezoidal area.
integabs	Absolute value of integral.
ise	Returns the integral of the squared deviation of the selected quantity from a target value that is entered via an additional argument.
itae	Returns the time-weighted squared deviation of the selected quantity from a target value that is entered via an additional argument.
itse	Returns the time-weighted squared deviation of the selected quantity from a target value that is entered via an additional argument. To use this function, you need to open the Add Trace Characteristics dialog and select the Error category.
j0	Bessel function of the first kind (0^{th} order).
j1	Bessel function of the first kind (1^{st} order).
jn	Bessel function of the first kind (nth order).
ln	Natural logarithm.
log	Natural logarithm (same as ln).
log10	Logarithm base 10.
lowercutoff	Returns the lower 3dB frequency of the selected simulation channel in Hertz.
lsidelobeX	The 'x' value for the left side lobe: the next highest value to the left of the max value.
lsidelobeY	The 'y' value for the left side lobe: the next highest value to the left of the max value.
mag	Magnitude of the complex number.
max	Returns maximum value of the simulation quantity.
max_swp	Returns maximum value of a sweep.
max2	Maximum value of the two simulation quantities. For example, **max2(a,b)** will plot maximum of **a** and **b** for a particular instance.
mean	Returns the average in the set of quantities selected. mean = sum(all y-value) / (number of y-values)

min	Returns the minimum value of the simulation quantity.
min_swp	Returns the minimum value of a sweep.
min2	Minimum value of the two simulation quantities. For example, **min2(a,b)** will plot minimum of **a** and **b** for a particular instance.
mod	Returns the modulus or absolute value of the simulation quantity.
nint	Nearest integer.
none	Returns null value.
normalize	Divides each value within a trace by the maximum value of the trace. ex. normalize(mag(x)).
odd	Returns 1 if integer part of the number is odd; returns 0 otherwise.
overshoot	Calculates peak overshoot given a threshold value and number of evenly spaced points over entire time range.
peakgain	Returns the peak value of gain of the selected simulation quantity in dB.
peakgainfreq	Returns the frequency in Hz at which the peak gain of the selected simulation quantity occurs.
polar	Coverts the complex number in rectangular co-ordinates to polar co-ordinates.
per	Returns the period of a simulation quantity.
phasecrossover	Returns the phase crossover frequency, at which the phase is -180 degrees, in Hz for the selected simulation quantity.
phasemargin	Returns the phase angle in degrees at the gain crossover frequency of the selected simulation quantity.
pk2pk	Peak to peak. Difference between max and min of the first parameter over the second parameter. Returns the peak-to-peak value for the selected simulation quantity.
pkavg	Returns the ratio of the peak to peak-to-average for the selected quantity.
pmax	Maximum period of the selected simulation quantity.
pmin	Minimum period of the selected simulation quantity.
pow	Raises x to the power of y; pow(x,y).
prms	Period Root Mean Square.
pulsefall9010	Returns the pulse fall time of the selected quantity according to the 90%-10% estimate.
pulsefront1090	Returns the pulse front time of the selected quantity according to the 10%-90% estimate.
pulsefront3090	Returns the pulse front time of the selected quantity according to the 30%-90% estimate.
pulsemax	Returns the pulse maximum from the front and tail estimates for the selected quantity.
pulsemaxtime	Returns the time at which the maximum pulse value of the selected

	quantity is reached.
pulsemin	Returns the pulse minimum from the front and tail estimates for the selected quantity.
pulsemintime	Returns the time at which the minimum pulse value of the selected quantity is reached.
pulsetail50	Returns the pulse tail time of the selected quantity from the virtual peak to 50%.
pulsewidth5050	Returns the pulse width of the selected quantity as measured from the 50% points on the pulse front and pulse tail.
pwl	Piecewise Linear.
pwl_periodic	Piecewise Linear for periodic extrapolation on x.
pwlx	Piecewise Linear x with linear extrapolation on x.
pw_minus	Pulse width of the first negative pulse.
pw_minus_avg	Returns the average of the negative pulse width input stream.
pw_minus_max	Returns the maximum pulse width of the negative pulse of input stream.
pw_minus_min	Returns the minimum pulse width of the negative pulse of input stream.
pw_minus_rms	RMS of the negative pulse width input stream.
pw_plus	Pulse width of the first positive pulse.
pw_plus_avg	Average of the positive pulse width input stream.
pw_plus_max	Max. Pulse width of the positive pulse of input stream.
pw_plus_min	Min. Pulse width of the positive pulse of input stream.
pw_plus_rms	RMS of the positive pulse width input stream.
re	Real part of the complex number.
rect	Converts the complex number in polar to rectangular co-ordinates.
rem	Fractional part of the selected simulation quantity (the remainder).
ripple	Returns the ripple factor (AC RMS/Mean) for the selected quantity.
risetime	Obtains the time taken to go from 10% to 90% of target point.
rms	Returnns the root mean square value of the selected quantity.
rmsAC	Returns the AC RMS for the selected quantity.
root	nth root function.
rSidelobeX	Returns the X value of right side-lobe occurrence.
rSidelobeY	Returns the Y value of right side-lobe occurrence.
settlingtime	Returns the latest time at which the value of the selected simulation quantity fell outside its tolerance band. The target value of the quantity and the +/- bandwidth of the tolerance band are the additional args.
sgn	Sign extraction.
sin	Sine.

sinh	Hyperbolic sine.
slidingmean	Returns the moving average value of the selected simulation quantity (specified by the first argument). The average is calculated over a period (specified by the second argument).
slidingrms	Returns the moving RMS value of the selected simulation quantity (specified by the first argument). The RMS value is calculated over a period (specified by the second argument).
sqr	Square of the selected simulation quantity.
sqrt	Square root of the selected simulation quantity.
stddev	Returns the standard deviation of given values.
sum	Returns the sum of the given values.
tan	Tangent.
tanh	Hyperbolic tangent.
undershoot	Calculates peak undershoot given a threshold value and number of evenly spaced points over entire time range.
uppercutoff	Returns the upper 3dB frequency of the selected simulation channel in Hz.
variance	Calculates the variance of the given values.
XAtYMax	Threshold crossing time: report first time (x value) at which an output quantity crosses YMax.
XAtYMin	Threshold crossing time: report first time (x value) at which an output quantity crosses a user definable threshold.
XAtYVal	Returns the X value at the first occurrence of Y value.
XWidthAtYVal	Returns the X width between the first 2 occurrence of Y value.
xdb10beamdwidth	Width between left and right occurrences of values 'x' db10 from max. Takes 'x' as argument (3.0 default). To use this function, you need to open the Add Trace Characteristics dialog and select the Radiation category.
xdb20beamwidth	Width between left and right occurrences of values 'x' db20 from max. Takes 'x' as argument (3.0 default). To use this function, you need to open the Add Trace Characteristics dialog and select the Radiation category.
YAtXMax	Returns the X value at maximum value of Y.
YAtXMin	Returns the Y value at minimum value of X.

Related Topics

Adding Trace Characteristics

Set Range Function

Selecting Solution Quantities to Plot

Each trace in a report includes a quantity that is plotted along an axis. The quantity being plotted can be a value that was calculated by Maxwell, such as L_{11}, or a value from a calculated expression.

To select a parameter to plot:

1. In the **Report** dialog box, select one of the solution type dependent categories. Some of the available categories and descriptions of their associated quantities are listed below. The actual list you see will depend upon the solution type and solution setup for your design. The selected category provides the default name for the plot, such as Force Plot *n*. You can edit the plot names in the project tree and the plot header in the report properties.

Variables	Intrinsic variables, such as frequency or theta, or user-defined project variables, such as the length of a quarter-wave transformer.
Output Variables	Derived quantities from the original field solution.
L, R, or Z	Inductance, Resistance, and Impedance matrix entries when the matrix parameter is defined.
Force or Torque	Force or Torque values when the parameter is defined.
Loss	▪ **Loss quantities for 2D and 3D transient solutions include:** CoreLoss, EddyCurrentLoss, ExcessLoss, HysteresisLoss, SolidLoss, StrandedLoss, StrandedLossR, and StrandedLossAC. [1] The availability of solution data for CoreLoss, EddyCurrentLoss, ExcessLoss, and HysteresisLoss depends on the material definitions in the design: ◦ If the design is hysteresis model-based, the solver provides solution data to plot only for HysteresisLoss, EddyCurrentLoss, and TotalLoss. ◦ If the design contains a non- hysteresis based model, core loss model type: Electrical Steel, the solver provides solution data to plot for all four quantities (CoreLoss, EddyCurrentLoss, ExcessLoss, and HysteresisLoss). ◦ For all other cases (for example: linear, power ferrite), the solver provides solution data to plot only for CoreLoss. **Note** For 2D transient solutions only, object-based quantities of CoreLoss, EddyCurrentLoss, ExcessLoss, and HysteresisLoss are also calculated, and can be selected for plotting, for objects whose **Core Loss Setting** has been checked in the **Set Core Loss** dialog box.. ▪ **Loss quantities for 2D and 3D eddy current solutions include:**

	CoreLoss, SolidLoss, StrandedLoss, StrandedLossAC, and StrandedLossR.
End Connection	• If a 2D transient End Connection excitation has been assigned, quantities include: Current, InducedVoltage, RingCurrent (the current in the first segment of the ring), and SolidLoss. • If a 2D eddy current End Connection excitation has been assigned, quantities include: Current and InducedVoltage.
Expression Cache	The values of expressions listed in the Expression cache of the Solution setup can be plotted, for example, as a function of adaptive pass to monitor their convergence.
Expression Converge	This can be used to plot convergence, as a function of adaptive pass, of expressions defined in the Expression cache of the Solution setup. When defining the report, for "Context", select Solution: *SetupN:Adaptive Pass*.

[1]*NOTE: StrandedLoss includes DC losses only; while StrandedLossAC includes DC losses and additional AC losses due to skin effect and proximity effect.*

2. Select a quantity to plot from the **Quantity** list. The available quantities depend on the selected category and the setup of the design.

Related Topics

Setting Core Loss for Transient and Eddy Current Solvers

Assigning an End Connection for a 2D Eddy Current Solver

Assigning an End Connection for a 2D Transient Solver

Selecting a Field Quantity to Plot

When plotting field quantities in the **Report** dialog box, the quantity can be a value that was calculated by Maxwell, such as a value from a calculated expression.

When you select **Fields** as the **Report Type** and select a line geometry from the **Solutions** pull-down list, three options are available for plotting versus length:

- Plot vs. actual distance along the line (in the units you specified).
- Plot vs. normalized distance.
- Plot vs. actual coordinate values of the line.

Plotting Imported Solution Data

1. In the **Solution** pull-down list in the **Report** dialog box, click the imported data you want to plot.
2. Follow the procedure for creating a report.

Post-Processed Quantities

Both pre-processing and post-processing quantities are available when creating reports.

For example, when you create a Magnetostatic Report for the **Magnetostatic Solver** the following are four of the quantities that appear in the **Category** list in the **Report** dialog box:

- **L** – The post-processed value of inductance (for example, the inductance of a 1-turn coil).
- **Lnom** – The inductance quantity that is solved with no post processing.
- **MagFlux** – The post-processed value of the total magnetic flux as seen by the coil (current source) including all sources (current, magnets, etc.).
- **MagFluxNom** – The total magnetic flux quantity that is solved with no post processing as seen by the coil (current source) including all sources (current, magnets, etc.)

Similarly, for an Eddy Current Report, the following are three of the quantities that appear in the **Category** list:

- **Winding** – If the design includes windings, Flux Linkage, Induced Voltage, Input Current, Input Voltage, and Current quantities are available (depending on the winding type). See also "Viewing Winding Results (Eddy Current)" on page 23-16 .
- **Loss** – Hysteresis Loss (Maxwell 2D Eddy current solutions only), Core Loss, Solid Loss, Stranded Loss Stranded Loss R, and Stranded Loss AC. See also "Viewing Loss Results (Eddy Current)" on page 23-17 .
- **End Connection** (Maxwell 2D Eddy current solutions only) – Induced Voltage, and Current. See also "Viewing End Connection Results (2D Eddy Current)" on page 23-16 .

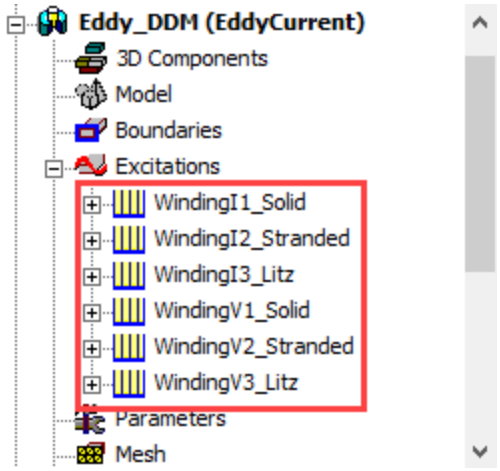
Per-Winding Loss Outputs for Maxwell 2D/3D Transient and Eddy Current Designs

Maxwell 2D/3D transient (T-omega only) and eddy current designs support loss quantity outputs for each winding when creating reports.

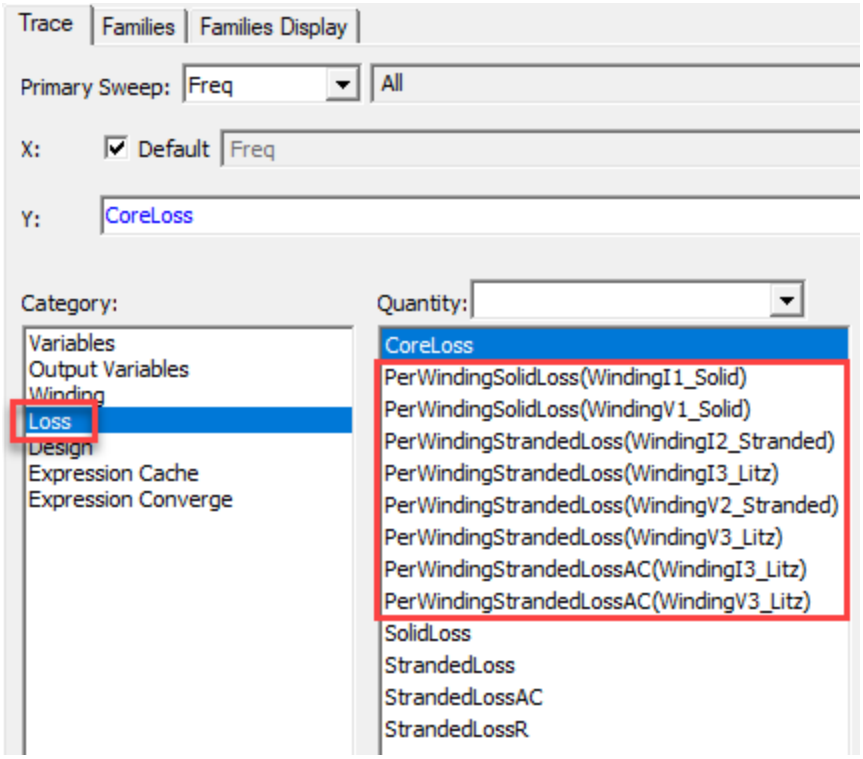
For the various winding types, the supported per-winding loss quantities are:

- Solid winding
  - **SolidLoss**
- Stranded winding without Litz wire
  - **StrandedLoss**
- Stranded winding with Litz wire
  - **StrandedLoss**
  - **StrandedLossAC**

For example, the winding assignments in the eddy current design shown below,



result in the following per-winding loss quantities available in the reporter.



> **Note:** Per-winding loss quantities will not affect legacy project reports.

Related Topics

Loss Quantities

Setting a Range Function

To apply a range function to the Y, Z, or Mag component of a trace:

Note	All range functions are available in the list regardless of solution context. You must select a function that is appropriate for the solution context and data type you are working with.

1. Click the **Range Function** button in the **Reports** dialog box.

   This opens the **Set Range Function** dialog. The functions available are the same as described in the Selecting a Function section, with the exception of those for the Eye Measurements category.

2. Click the Specified radio button on the Range function line.

   This enables the Range Function fields.

3. Select the **Category**, and then an associated Function to apply. The available categories depend on the plot, and Category enables the display of associated functions.

Category	Functions for the Category
Math	max, min, pk2pk, rms, integ, integabs, avgabs, rmsAC, ripple, pkavg, XatYMin, XatYMax, XatYVal
PulseWidth	pulsefall9010, pulsefront9010, pulsefront3090, pulsemax, pulsemaxtime, pulsemin, pulsemintime, pulsetail50, pulsewidth5050, pw_plus, pw_plus_max, pw_plus_min, pw_plus_avg, pw_plus_rms, pw_minus_max, pw_minus_min, pw_minus_avg, pw_minus_rms
Overshoot, Undershoot	overshoot, undershoot.
TR & DC	crestfactor, formfactor, distortion, fundamentalmag, delaytime, risetime, deadtime, settlingtime,
Error	iae, ise, itae, itse
Period	per, pmax, pmin, prms
Radiation	xdb10bandwidth, xdb20bandwidth, lSidelobeX, lSidelobeY, rSidelobeX, rSidelobeY
Eye Measurements	EyeLevelZero, EyeLevelOne, EyeAmplitude, EyeHeight, EyeSignalToNoise, EyeOpeningFactor, EyeWidth, EyeJitterP2P, EyeJitterRMS, EyeRiseTime, EyeFallTime

   Given a selected Function, and Category, the **Set Range Function** dialog displays a text field that explains the Purpose of the function. For a full list of functions and their definitions, see Selecting a Function.

   Selecting a function causes the display of a description in the **Purpose** field. If the function requires a value (such as the XatYVal Math function or the pw_minus_max Pulse Width

function), the following table the function field displays the name, editable value field, unit, and description.

4. Use the **Over Sweep** drop down menu to select from available sweeps.
5. To select from available Sweeps, or to edit them, click the ellipsis [**...**] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time sweeps). By default, **Use all values** is selected.
   - For sweeps other than Time, unchecking **Use all values** enables a list of available sweep values. You can use the buttons to **Clear All** or **Select All** sweep values.
     a. Select the Sweeps **Default** or **Edited** radio buttons to specify whether to accept the default or edited sweeps.
     b. To edit the sweeps further, select the ellipsis [**...**] button to display an **Edit Sweep** dialog box. For frequency variables, this lets you specify a single value, linear step, linear count, decade count, octave count, or exponential count. You can **Add** legal values to the list of sweep values, **Update** the list for changes, or **Delete** selected entries.
     c. When finished selecting/editing sweep values, close the dialog box. The sweep values you select are displayed on the Over Sweep line.
   - For Time sweeps:
     a. Select the **Specify range** radio button to enable the **Min** and **Max** value fields.
     b. Enter the desired **Min** and **Max** values, and choose the associated units of measure from the drop down lists.
     c. When finished, close the dialog box. The sweep values you select are displayed on the Over Sweep line.
6. Click **OK** to apply the range function.

Related Topics

Selecting a Function

Eye Measurement Range Function Parameters

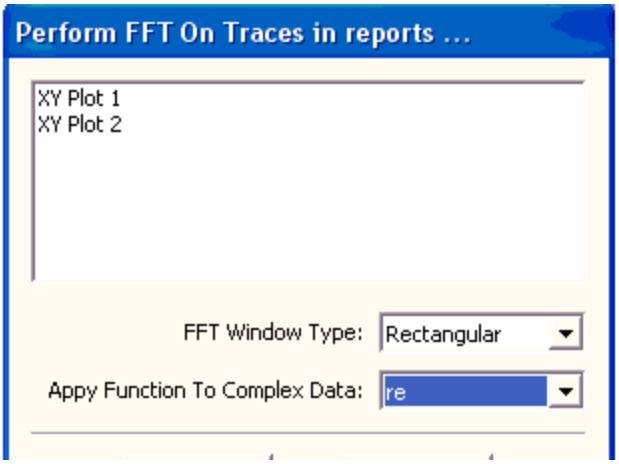
The Eye Measurement category of range functions provide the means to evaluate several characteristics of eye diagrams. Each of the Eye Measurement functions includes the following parameters. Specify the Value by editing the Value text field for the parameter name. Specify the unit for the parameter by selecting from the Unit drop down menu.

Name	Default Value	Default Unit	Description
Unit Interval	0	ns	Unit interval of signal
Start Offset	0	ns	Offset at beginning of signal
End Offset	0	ns	Offset at end of signal
Autocrossing	1		Nonzero number means that crossing amplitude is

Name	Default Value	Default Unit	Description
amplitude			calculated automatically.
Crossing amplitude	0	mV	Specify crossing amplitude used for eye measurement data computation.

Perform FFT on a Report

You can perform FFT on an existing 2D plot by using the **Results>Perform FFT** command. This opens the **Perform FFT on Traces in Reports** dialog.



1. Select the report you want from the list in the dialog.
2. Select the FFT Window type from a drop down list.

   Select the window type to apply. Windowing functions cause the FFT of the signal to have non-zero values away from ω. Each window function trades off the ability to resolve comparable signals and frequencies versus the ability to resolve signals of different strengths and frequencies.
3. Select the function to apply to complex data.

The new report displays and appears in the Project tree. The new report name prefixes FFT to the name of the original report. Trace names are also prefixed with FFT.

Related Topics

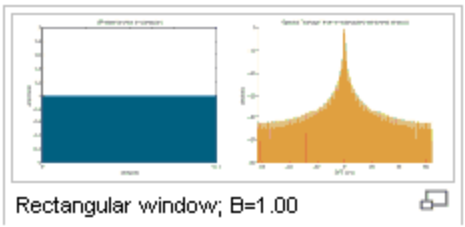
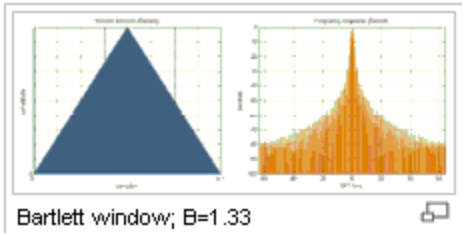
FFT Window Functions

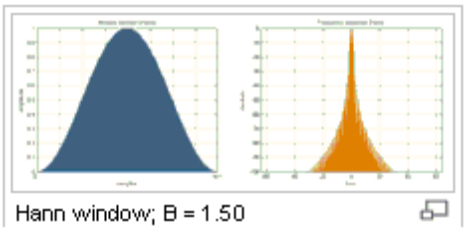
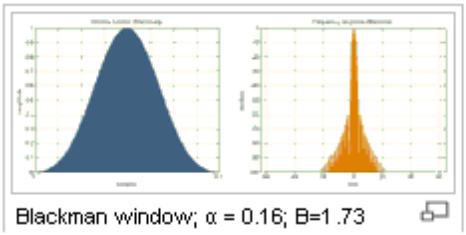
Apply FFT to Report Functions

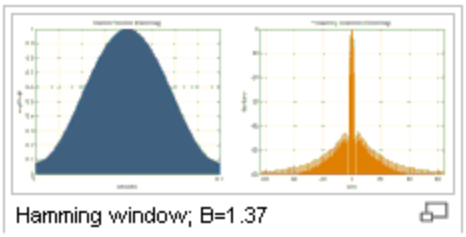
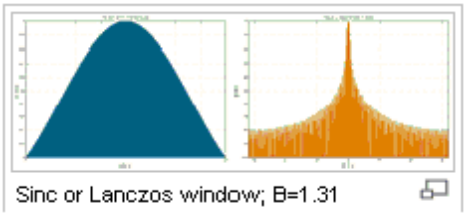
Perform TDR on a Report

FFT Window Functions

The window type list for **Perform FFT on Report** includes:

Window Function	Preferred Use
Rectangular	A low dynamic range function offering good resolution for signals of comparable strength. Poor when signals have very different amplitudes. $w(n)=1$. 
Tri	A Bartlett window with the endpoints valued at zero. 
Van Hann	A moderate dynamic range function, designed for narrow band applications. $$w(n) = 0.5\left(1-\cos\frac{2\pi n}{N-1}\right)$$

Window Function	Preferred Use
	 Hann window; B = 1.50
Blackman	A high dynamic range function, with lower resolution, designed for wide band applications. $w(n) = a_0 - a_1 \cos\left(\frac{2\pi n}{N-1}\right) + a_2 \cos\left(\frac{4\pi n}{N-1}\right)$ where $a_0 = (1-\alpha)/2$; $\alpha_1 = 1/2$; $\alpha_2 = \alpha/2$  Blackman window; α = 0.16; B=1.73

Window Function	Preferred Use
Hamming	A moderate dynamic range function, designed for narrow band applications. It minimizes the maximum sidelobe. $$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)$$  Hamming window; B=1.37
Lanczos	The Lanczos window offers a windowed form of the infinite sinc filter, providing the central lobe of a horizontally-stretched sinc, sinc(x/a) for -a £ x £ a.  Sinc or Lanczos window; B=1.31
Weber	
Welch	This approach applies a parabola-shaped window to the frequency domain data. It is based on the Bartlett method but splits the signal into overlapping segments, which are then windowed. The intent is to balance the influence of data in the center of the function.

Related Topics

Perform FFT on a Report

FFT Window Functions
Apply FFT to Report Functions
Perform TDR on a Report

Apply FFT to Report Functions

The choices include:

ang_deg	Angle (phase) of a complex number, cut at +/-180
ang_rad	Angle in radians
arg	
cang_deg	Cumulative angle (phase) of the first parameter (a complex number) in degrees, along the second parameter (typically sweep variable). Returns a double precision value cut at +/-180.
cang_rad	Cumulative angle of the first parameter in radians along a second parameter (typically a sweep variable) Returns a double precision value.
dB(x)	20*log10(|x|)
dB 10normalize	10*log [normalize(mag(x))]
dB 20normalize	20*log [normalize(mag(x))]
dBc	
im	Imaginary part of the complex number
mag	Magnitude of the complex number
normalize	Divides each value within a trace by the maximum value of the trace. ex. normalize(mag(x))
re	Real part of the complex number

Related Topics

FFT Window Functions
Perform FFT on a Report
Perform TDR on a Report

Perform TDR on Report

The idea behind Time-Domain Reflectometry (TDR) is to excite a structure with a step or impulse function, and inspect the reflections as a function of time. Before you can examine the time domain, you must perform a sweep for a transient or eddy current solution. You can perform TDR

on an existing 2D plot by using the **Results>Perform TDR on Report** command. This opens a **Perform TDR On Traces in reports...** dialog.

1. Select the report you want from the list in the dialog.
2. Specify the input signal as Step or Impulse and give the rise time.
3. Select the TDR Window type from a drop down list.

   Select the window type to apply. Windowing functions cause the FFT of the signal to have non-zero values away from ω. Each window function trades off the ability to resolve comparable signals and frequencies versus the ability to resolve signals of different strengths and frequencies.

   You may choose to specify a window width as a percentage.

   If you select the Kaiser function, you can specify a Kaiser number.

The new report displays and appears in the Project tree. The new report name prefixes TDR to the name of the original report. Trace names are also prefixed with TDR.

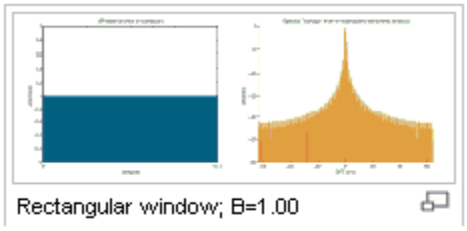
Related Topics

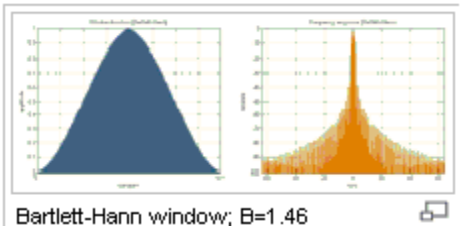
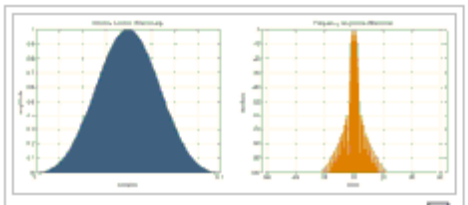
Perform FFT on a Report

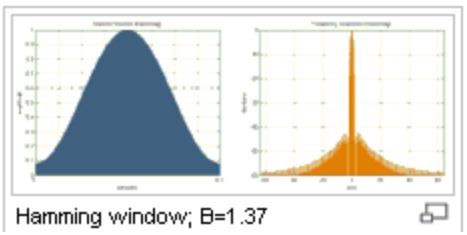
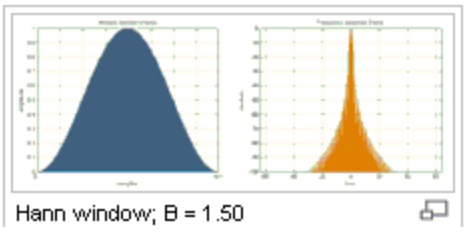
Time Domain Plots in Eddy (IFFT)

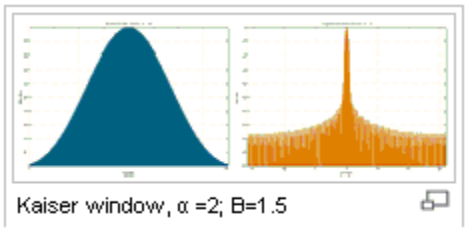
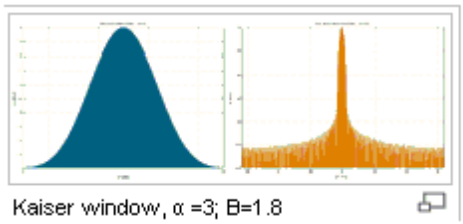
TDR Windowing Functions

Windowing functions cause the FFT of the signal to have non-zero values away from ω. Each window function trades off the ability to resolve comparable signals and frequencies versus the ability to resolve signals of different strengths and frequencies. The window type list includes:

Window Function	Preferred Use
Rectangular	A low dynamic range function offering good resolution for signals of comparable strength. Poor when signals have very different amplitudes. $w(n)=1$. 
Bartlett	A high dynamic range function, with lower resolution, designed for wide band

Window Function	Preferred Use
	applications. $$w(n) = a_0 - a_1\left|\frac{n}{N-1} - \frac{1}{2}\right| - a_2\cos\left(\frac{2\pi n}{N-1}\right)$$ where a_0=0.62; a_1=0.48; a_2=0.38  Bartlett-Hann window; B=1.46
Blackman	A high dynamic range function, with lower resolution, designed for wide band applications. $$w(n) = a_0 - a_1\cos\left(\frac{2\pi n}{N-1}\right) + a_2\cos\left(\frac{4\pi n}{N-1}\right)$$ where a_0 =(1-α)/2; α_1=1/2; α_2=α/2  Blackman window; α = 0.16; B=1.73

Window Function	Preferred Use
Hamming	A moderate dynamic range function, designed for narrow band applications. It minimizes the maximum sidelobe. $$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)$$  Hamming window; B=1.37
Hanning (default)	A moderate dynamic range function, designed for narrow band applications. $$w(n) = 0.5\left(1-\cos\frac{2\pi n}{N-1}\right)$$  Hann window; B = 1.50
Kaiser	Selecting the Kaiser plot also enables a field to specify an associated Kaiser parameter. The larger the Kaiser parameter, the wider the window. The

Window Function	Preferred Use
	parameter controls the trade off between width of the central lobe and the area of the side lobes.  Kaiser window, α =2; B=1.5  Kaiser window, α =3; B=1.8
Welch	This approach applies a parabola-shaped window to the frequency domain data. It is based on the Bartlett method but splits the signal into overlapping segments, which are then windowed. The intent is to balance the influence of data in the center of the function.

You can use the **Save as Default** to set the current values as a default, and the **Use Defaults**

Spectral Domain Plots in Transient (FFT)

When plotting in the spectral domain, you must specify the parameters for the Fast Fourier Transform (FFT). The FFT function used by Maxwell uses windowing; therefore, the user must enter a start and stop time within which the data is periodic.

To enter the FFT parameters:

1. In the **Report** window, be sure the **Domain** is set to **Spectral**.
2. In the **Context** section of the reporter dialog box, click **FFT Options.**

   The **FFT Options** dialog appears.
3. Enter the **Start Time** for the window using the pull-down menu.
4. Enter the **End Time** using the pull-down menu.
5. Enter the **Window Type** using the pull-down menu.

The Samples and number of points are calculated based on the chosen start and end values. The signal must be periodic during the time interval chosen or an error will result.

Related Topics

Time Domain Plots in Eddy (IFFT)

Time Domain Plots in Eddy (IFFT)

When creating time domain plots from frequency domain data, you must specify the parameters for the Inverse Fast Fourier Transform (IFFT):

1. In the **Report** window, be sure the **Domain** is set to **Time**.
2. In the **Context** section of the reporter dialog box, click **IFFT Options.**

   The **IFFT Options** dialog appears.
3. In the **Input Signal** section of the dialog select either **Impulse** or **Step** signal to use in performing the IFFT.
4. For **Step** signal, input the rise time for the pulse. The rise time should chosen so that the frequency content is within the bandwidth of the available frequency sweep data.
5. Select the **Window Type** from the pull-down menu.
6. Enter the window **Width** in percent.

The **Maximum Plot Time** and the **Delta Time** are calculated based on the solution data available.

Related Topics

Spectral Domain Plots in Transient (FFT)

Specifying Output Variables

You can access the **Output Variables** window in several ways.

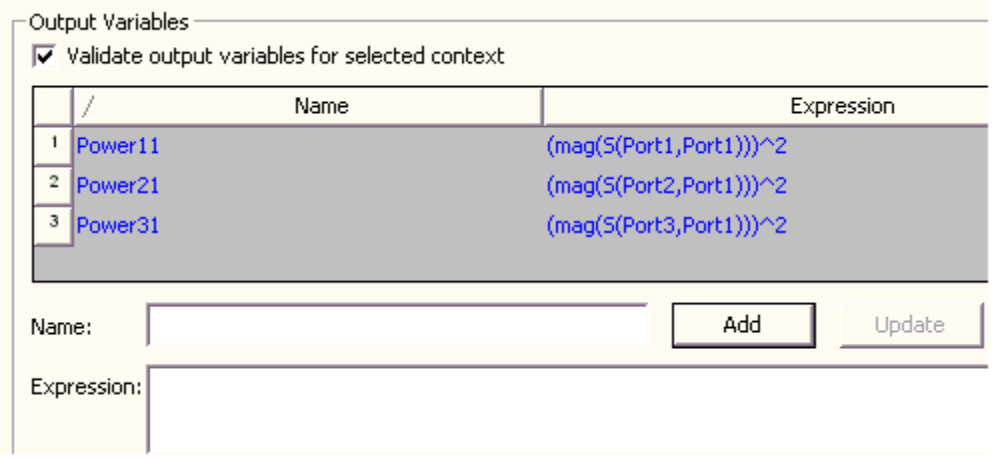
- Click **Maxwell3D, Maxwell2D,** or **RMxprt** and then select **Results>Output Variables**.
- In the Project tree, right-click on **Results** and select **Output Variables** from the short-cut menu.
- In the **Solution Setup** dialog box, select the **Expression Cache** tab, click the **Add...** button to display the **Add to expression cache** dialog box, and click the **Output Variables** button.
- Click the **Output Variables** button the **Reports** dialog.

The **Output Variables** window contains four sections:

- **Context section,** where you specify the Report type, the Solution, and for appropriate report types, the Domain. Changing the Report type affects whether the Domain menu appears, and may affect the functions listed in the Calculation section.

- **Output Variables** section, where you can specify the name and expression for a new output variable. Existing variables appear in the list at the top of the window. Clicking the triangle in the Name bar inverts the sort order.

  At top of the **Output Variables** window, you can use a check box to **Validate output variable for selected context**.



- **Calculation section**, where you can insert quantities into the Expression area of the **Output Variables** section.
- **Function section**, where you can insert completed expressions into the Expression area of the **Output Variables** section.

Related Topics

Adding a New Output Variable

Building an Expression Using Existing Quantities

Viewing the Output Variable Convergence

Adding a New Output Variable

To add an output variable:

1. Click **Maxwell3D, Maxwell2D,** or **RMxprt** and then select **Results>Output Variables** or, in the Project tree, right-click on **Results** and select **Output Variables** from the short-cut menu, or on the **Modify Report** dialog box, click the Output Variables button.

   The **Output Variables** window appears. Variables defined using the **Results>Output Variables** command appear in the list at the top of the window.
2. In the **Output Variables** section, enter a name for the new variable in the **Name** box.
3. To enter an expression, do one or both of the following:
   a. Type part or all of the expression directly in the **Expression** area. Valid functions appear in blue. Invalid functions appear in red.

b. Insert part or all of the expression using the options in the Calculation and Function sections.

4. Optionally, choose to **Validate output variables for selected context**.
5. Click **Add** to add the new variable to the list.
6. Repeat steps 2 through 4 to add additional variables.
7. When you are finished adding output variables, click **Done** to close the **Output Variables** window.

Related Topics

Deleting Output Variables

Building an Expression Using Existing Quantities

Viewing the Output Variable Convergence

Building an Expression Using Existing Quantities

When you are entering an expression for a new output variable, you can insert part or all of the expression using the options in the **Calculation** and **Function** sections of the **Output Variables** window.

To add an input variable by inserting part or all of the expression:

1. Click **Maxwell3D, Maxwell2D,** or **RMxprt** and then select **Results>Output Variables** or, in the Project Tree, right-click on **Results** and select **Output Variables** from the short-cut menu, or on the **Modify Report** dialog box, click the Output Variables button.

   The **Output Variables** window appears.
2. In the **Output Variables** section, enter a name for the new variable in the **Name** box.
3. To insert a quantity:
   a. From the **Report Type** pull-down list, select the type of report from which you want to select the quantity.
   b. From the **Solution** pull-down list, select the solution from which you want to select the quantity.
   c. From the **Category** list, select the type of quantity you want to enter.
   d. From the **Quantity** list, select the quantity or the geometry.
   e. From the **Function** list, select a ready-made function (this option is the same as inserting the function from the **Function** section).
   f. If applicable, from the **Domain** list, select the solution domain.
   g. Click **Insert Into Expression**.
   The selected quantity is entered into the **Expression** area of the **Output Variables** section.
4. To insert a function:
   a. In the **Function** section, select a ready-made function from the pull-down list.
   b. Click **Insert Function into Expression**.
   The function appears in the **Expression** area of the **Output Variables** section.

5. When you are finished defining the variable in the **Expression** area, click **Add** to add the new variable to the list.
6. Repeat steps 2 through 5 to add additional variables.
7. When you are finished adding output variables, click **Done** to close the **Output Variables** window.

Note	Remember the evaluated value of an expression is always interpreted as in SI units. However, when a quantity is plotted in a report, you have the option to plot values in units other than SI.

Related Topics

Adding a New Output Variable

Building an Expression Using Existing Quantities

Function List for Output Variables

The Output Variables dialog includes a second function list containing functions to enter directly into the Expression field. These functions can also be applied to previously specified Quantities and Functions.

Some of these functions can operate along an entire curve. These are: deriv, min, max, integ, rms, pk2pk, cang_deg and cang_rad.

You can select from the following functions in the **Output Variables** dialog **Function** list, or type them directly into the Expression field, if necessary:

abs	Absolute value
acos	Arc cosine
acosh	Hyperbolic arc cosine
ang_deg	Angle (phase) of a complex number, cut at +/-180
ang_rad	Angle in radians
arg	Argument of a complex number. It is the angle the complex number makes with the positive x axis. Same as **ang_deg**.
asin	Arc sine
asinh	Hyperbolic arc sine
atan	Arc tangent. Takes a tangent value as an argument. Because there are two angles in a circle that can have the same tangent value, and atan can return only one value, it returns a value in the range between -90 degrees and +90 degrees (or between -pi/2 and pi/2 in radians).
atan2(y,x)	A two-argument version of the **atan** function. Takes the y and x coordinates (including sign information) of a point as arguments and returns the angle from the X-axis. Can return angle values for the full circle (-180 degrees to +180 degrees or -pi to +pi in radians).

atanh	Hyperbolic arc tangent
avgabs	Absolute value of average.
cang_deg	Cumulative angle (phase) of the first parameter (a complex number) in degrees, along the second parameter (typically sweep variable). Returns a double precision value cut at +/-180.
cang_rad	Cumulative angle of the first parameter in radians along a second parameter (typically a sweep variable) Returns a double precision value.
cmplx(*re*, *im*)	A complex number, where *re* is the real part and *im* is the imaginary part.
conjg	Conjugate of the complex number.
cos	Cosine
cosh	Hyperbolic cosine
crestfactor	Peak/RMS (root mean square) for the selected simulation quantity
dB(x)	20*log10(|x|)
dB10	10*log(|x|) to base 10.
dBm(x)	10*log10(|x|) +30
dBW(x)	10*log10(|x|)
dBc	Decibels relative to the carrier. It is the power ratio of the signal to a carrier signal. Gives the relative signal strength.
db10normalize	10*log [normalize(mag(x))]
db20normalize	20*log [normalize(mag(x))]
degel	Conversion from degrees electrical to seconds with respect to Hz.
deriv	Derivative of a given parameter.
even	Returns 1 if integer part of the number is even; returns 0 otherwise
exp	Exponential function (the natural anti-logarithm)
if	if(cond_exp,true_exp, false_exp).
im	Imaginary part of the complex number
int	Truncated integer function
j0	Bessel function of the first kind (0^{th} order)
j1	Bessel function of the first kind (1^{st} order)
ln	Natural logarithm
log	Natural logarithm (same as ln).
log10	Logarithm base 10
mag	Magnitude of the complex number
mod	Returns the modulus or absolute value of the simulation quantity.
nint	Nearest integer
normalize	Divides each value within a trace by the maximum value of the trace. ex.

	normalize(mag(x))
odd	Returns 1 if integer part of the number is odd; returns 0 otherwise
polar	Converts the complex number in rectangular to polar
pow	Raise x to y power., pow(x,y).
pwl	Piecewise Linear
pwl_periodic	Piecewise Linear for periodic extrapolation on x
pwlx	Piecewise Linear x with linear extrapolation on x.
re	Real part of the complex number
rect	Converts the complex number in polar to rectangular
rem	Returns the fractional part of a decimal number such that rem(x) = x-int(x) Syntax: rem(x)
root	Take the y root of x, root(x,y)
sgn	Sign extraction
sin	Sine
sinh	Hyperbolic sine
sliding mean	sliding mean.
slidingrms	sliding root mean square
sqr	Square.
sqrt	Square root
tan	Tangent
tanh	Hyperbolic tangent
y0	Bessel function of the second kind (0^{th} order)
y1	Bessel function of the second kind (1^{st} order)
yn	Bessel function of the second kind (nth order)

Related Topics

Adding Trace Characteristics

Set Range Function

Deleting Output Variables

To delete output variables:

1. Remove all references to the output variable in the project.
2. Save the project to erase the command history.

3. Click **Maxwell3D**, **Maxwell2D** or **RMxprt**, and then select **Results>Output Variables** or, in the Project Tree, right-click on **Results** and select **Output Variables** from the short-cut menu, or on the **Modify Report** dialog box, click the **Output Variables** button.

   This opens the **Output Variables** dialog.
4. Select the variable and click the **Delete** button.
5. Click **OK** to close the dialog.

Related Topics

Adding a New Output Variable

Report Data

The follow sections describe how you can view, update, export, import, override, and delete various values used for display in plots and reports.

The topics for this section include:

Updating Reports (Post-Processing Data)

Deleting Reports

Exporting Plot Data

Importing 2D Plot Data

Report File Formats

Exporting Graphics Files from a Plot

Plotting Imported Solution Data

Setting a Range Function

Updating Reports (Post-Processing Data)

To manually update the report that is active in the design area using the latest simulation results, right-click in the report and select **Update Report** from the menu. Alternatively, you can right-click on the report under **Results** in the Project Manager and select **Update Report** from the context menu.

If you wish to update all reports in the design, select **Maxwell 2D** (or **Maxwell 3D**)**>Results>Update All Reports**. Alternatively, right-click on the **Results** icon in the Project Manager and select **Update All Reports** from the menu.

Deleting Reports

- To delete a single report, right-click on the report in the **Results** section of the **Project Manager** window for the design, and click **Delete** on the menu.

Warning	Solution data that have been deleted cannot be

	recovered!

- To delete all reports, either select **Maxwell 2D** (or **Maxwell 3D** or **RMxprt**)**>Results>Delete All Reports**, or right-click on the **Results** icon in the **Project Manager** window for the design, and click **Delete All Reports** on the pop-up menu.

Warning	Deletions occur as soon as you click this button and cannot be undone. If you change your mind while in this window, click **Cancel** BEFORE clicking **Do Deletions**.

Opening Reports

- To open a single report, right-click on the report in the **Results** section of the **Project Manager** window for the design, and click **Open Report** on the menu.
- To open all reports, either select **Maxwell 2D** (or **Maxwell 3D**)**>Results>Open All Reports**, or right-click on the **Results** icon in the **Project Manager** window for the design, and click **Open All Reports** on the pop-up menu.

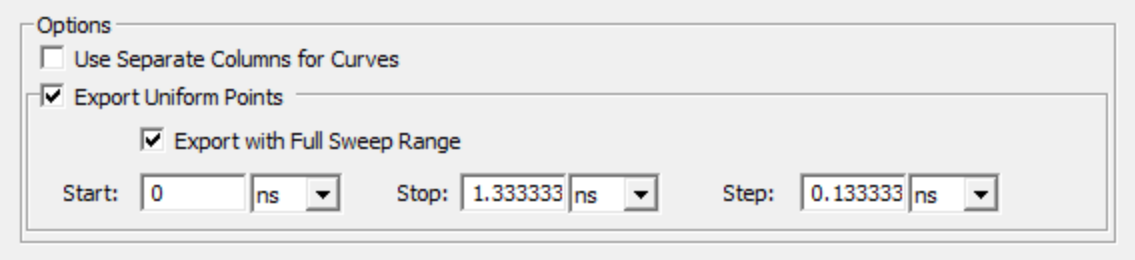
Exporting Reports

You can export reports in a variety of text and graphic formats. You must have an existing plot open to see the corresponding Report2D or Report3D menu.

1. Click **Report2D** or **Report 3D** on the top menu, and then select **Export**. Alternatively, right-click on the plot, and then click **Export ...** on the shortcut menu.

   The **Export Report** dialog box opens.
2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** box.
4. Select one of the following file formats from the **Save as type** pull-down list:
   - ***.csv** – Comma delimited data files.
   - ***.tab** – Tab delimited data files.
   - ***.dat** – Ansoft PlotData files (2D reports only).
   - ***.txt** – Post processor format files (2D reports only).
   - ****.exy*** – Ansys EnSight XY date files. (2D reports only).
   - ***.case** – Ansys case files. (3D reports only).
   - ***.rdat** – Ansoft Report Data format files.
   - ***.emf** – Microsoft EMF files.
   - ***.gif** – Graphics Interchange Format files.
   - ***.bmp** – Bitmap files.
   - ***.wrl** – Virtual Reality Modeling Language (VRML) files.
   - ***.tif, *.tiff** – Tagged Image File Format files.
   - ***.jpg, *.jpeg** – Joint Photographics Experts Group files.

5. Optionally, for .csv and .tab files, you can check, Use Separate Columns for curves.
6. Optionally, for 2D reports, you can choose to Export Uniform Points, which enables fields in which you can specify the start, stop, and step values and units to output uniformly spaced point data.



7. Click **Save** to export the report to the specified location in the selected data or graphical format.

Related Topics

Importing 2D Plot Data
Creating a Report from a Report Data File
Report File Formats

Importing 2D Plot Data

To import 2D plot data to a report:

1. Left-click **Report2D** on the top menu, and then select **Import Data**.
   The **Open** dialog box opens.
2. Browse to the directory where the data file is located.
3. Select the file format from the **Files of type** drop-down. Options include:
   - ***.csv** – Comma delimited data files.
   - ***.tab** – Tab delimited data files.
   - ***.dat** – Ansoft PlotData files.
   - ***.txt** – Post processor format files.
   - ***.rdat** – Ansoft Report Data files.

   For more information, see Report File Formats.
1. Select the filename.
2. Click **Open** to import and plot the data, or click **Cancel** to close the dialog box without importing any data.

Related Topics

Exporting Reports

Creating a Report from a Report Data File
Report File Formats

Report File Formats

The following table provides information about the file formats that can be imported into plot data reports:

File Extension	Information
csv	• Uses a comma (,) as the separation character. • The x-axis value is in the first column. Each curve's Y-values make up one column, and the curves are in the same order as in the plot legend. • The first row is the X-axis name [unit] and the information for each curve, which is the same as the plot curve legend. • Other rows are the X-axis value and the Y-value for each curve.
tab	• Uses the tab character as the separation character. • The x-axis value is in the first column. Each curve's Y-values make up one column, and the curves are in the same order as in the plot legend. • The first row is the X-axis name [unit] and the information for each curve, which is the same as the plot curve legend. • Other rows are the X-axis value and the Y-value for each curve.
txt	• The txt file must begin with a header that contains the following: • A first line made of 92 equal signs (=) • A second line that begins with the company name and ends with the date in MM/DD/YY format • A third line that begins with a plot name and ends with the time in hh:mm:ss format • A fourth line, which is empty • A fifth line made of 92 hyphens (-) **Note:** Each header line must be 92 characters in length, except the empty line. • Each column has a fixed width, separated by white space. • The x-axis value is in the first column. Each curve's Y-values make up one column, and the curves are in the same order as in the plot legend. • The first row is the X-axis name [unit] and the trace name [unit] for each curve, which is the same as the plot curve legend. • The second row is an empty column and the variable values for each curve, which is the same as the plot curve legend. • Other rows are the X-axis value and the Y-value for each curve.
dat	DAT is an Ansys-specific format. Ansys recommends that you do not generate files using this format. Files imported with these formats should be exported only

File Extension	Information
	from Ansys products.
rdat	RDAT is an Ansys-specific format. Ansys recommends that you do not generate files using this format. Files imported with these formats should be exported only from Ansys products.

Related Topics

Importing 2D Plot Data

Exporting Reports

Creating Custom Report Templates and Defaults

You can edit properties from any report type and save it as a template or as the default. This can save repeated editing of properties (for example, the company name, or color schemes) when you create other reports. Once you create templates, you can access them from the **Results>Report Templates>** menu and the **2D Reports> Report Templates>Apply Settings** menu.

See Modifying the Background Properties of a Report for a discussion of format changes you can make to any report.

To save an edited report as a template:

1. In the Project Tree, right-click on the report name of interest to display the shortcut menu and click **Report Templates>Save**. You can also click **Report2D>Report templates>Save**.

   This displays the **Report Save As Report Template** file browser. By default, the directory is your AnsysEM\<*productName*>\userlib\ReportTemplates directory. You can also save to the SysLib directory.
2. Typically, you accept the directory.
3. You must provide a file name, which will be given an .rpt extension.

   It is good practice to give the template a descriptive name, showing both the kind of format you begin with (such as XY Plot or 3D Plot) and apt description of the distinguishing edits (such as for company name, or color scheme). Once saved, this name will appear on the **PersonalLib** menu.
4. The **Save As Typ**e field currently supports the Ansoft Report Format (.rpt) format.
5. Click **Save** to save the template to the PersonalLib menu.

All .rpt templates in the directory appear on the **Results>Report Templates>PersonalLib** menu. Selecting a report from the PersonalLib menu opens a report that you can then Modify to add traces or perform other edits. Templates in the SysLib directory appear on the **Report Templates** menu.

To save an edited report as a default:

1. In the Project Tree, right-click on the report name of interest to display the shortcut menu and click **Report Templates>Save Settings as default**. You can also click **Report2D>Report templates>Save Settings as default** or **Report3D>Save Settings as default**.

Related Topics

Modifying Reports
Setting Report2D options
Modifying the Background Properties of a Report
Modifying the Legend in a Report
Working with Traces
Editing the Display Properties of Traces
Discarding Report Values Below a Specified Threshold

Exporting Reports as Graphics

You can export reports as figures in several formats. You must have an existing plot open to see the Report2D or Report 3D menu.

To export a graphics file:

1. On the **Report 2D** or **Report3D** menu, click **Export**.

   The **Export Report** dialog box opens:
2. Click the **Browse...** button to open the Export Report browser window.
3. Specify the file location and name, and select a graphics format from the dropdown list.
4. Select the desired graphics file format from the **Save as type** drop-down list.

- ***.jpg** – Joint Photographics Experts Group files.
- ***.gif** – Graphics Interchange Format files.
- ***.bmp** – Bitmap files.
- ***.tiff** – Tagged Image File Format files.
- ***.wrl** – Virtual Reality Modeling Language (VRML) files.

5. Click **Save**. The file is exported to the specified location as a graphics file.

To export an image file of a report to specified high resolution:

1. Set platform environment variables for the desired width and height values, (for example, for 8k, use width var = 7680 and height var=4320): ANSYS_EM_EXPORTED_IMAGE_WIDTH and ANSYS_EM_EXPORTED_IMAGE_HEIGHT.
2. Click With a report open, click **Report2D**>**Export...** or **Report3D**> **Export...**and choose an image format like .jpeg, and Export.

   The image created at the specified resolution. If you record a script for the image export, the parameters for the width and height will match those specified by the system

environment variables. Fonts and line thickness is not scaled. Only the image. You will have to iteratively increase font sizes till you find a suitable output.

Related Topics

Exporting Reports

Copying Reports to the Clipboard as Images

You can copy images of reports to the clipboard, so that they can be pasted into other applications as follows:

1. Make the report the active window. Alternatively, for a plot-on-schematic, click on the plot to select it.
2. Click **Edit>Copy Image**, or right-click on the report window or plot-on-schematic and select **Copy Image** on the context menu to copy the report to the clipboard as an image.
3. Open the application into which you want to paste the image, and paste the image.

Related Topics

Copy and Paste of Report and Trace Data

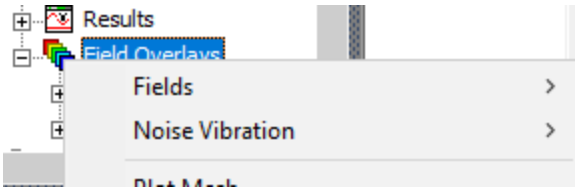
Copy and Paste of Report and Trace Definitions

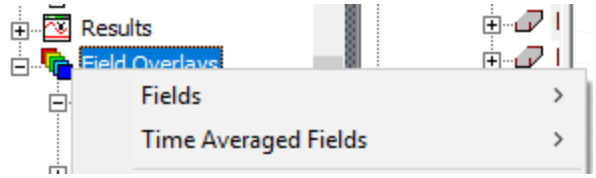
Plotting Field Overlays

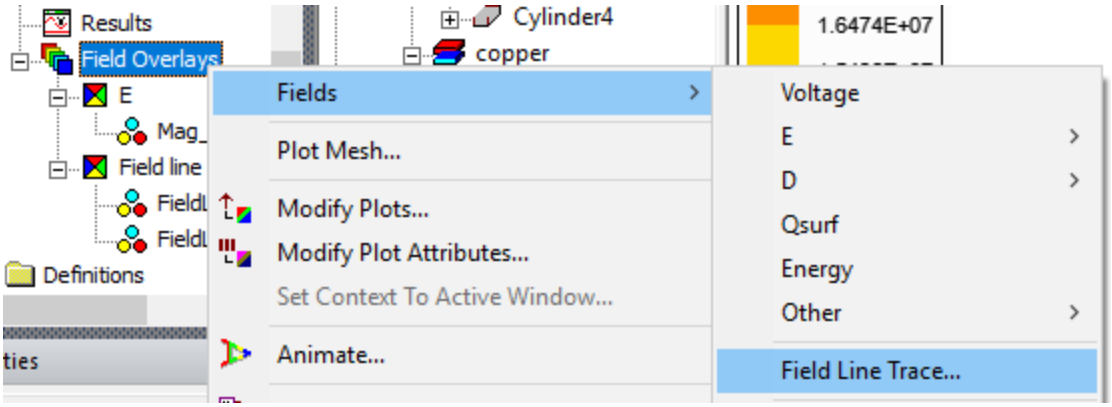
Field overlays are representations of basic or derived field quantities on surfaces or objects for the current design variation. You must select a geometry to create a plot. You can set the design variation via the **Set Design Variation** dialog box. This dialog box is accessible from the **Solution Data** window via by clicking the ellipsis button on the right of the Design Variation field, and via the **Results>Apply Solved Variation** command.

You can control the visibility of Field Overlays by means of the **View>Visibility>Active View Visibility** dialog box, **Fields Reporter** tab, shown above or by right clicking on the Field plot in the Project tree and checking or unchecking **Plot Visibility**.

There are several types of Field overlays that you can plot:







- **Fields**
- **Time Averaged Fields** (*Transient designs only*)
- **Noise Vibration** *(Transient designs if element-based harmonic forces are set up)*
- **Field Line Trace** (*Electrostatic designs only*)

Related Topics

Plotting Standard Fields Overlays
Plotting Noise Vibration Overlays
Plotting Field Line Trace Overlays

Plotting Standard Fields Overlays

Field overlays are representations of basic or derived field quantities on surfaces or objects for the current design variation. You can set the design variation via the **Set Design Variation** dialog box. This dialog box is accessible from the **Solution Data** window via by clicking the ellipsis button on the right of the Design Variation field, and via the **Results>Apply Solved Variation** command.

To plot a standard field quantity:

1. Select a point, line, surface, or object to create the plot on or within. You may also select a plane or object list in the **History Tree**.

> **Note:** For 2D Designs, a plane selection must be consistent with the drawing plane or an error will result.

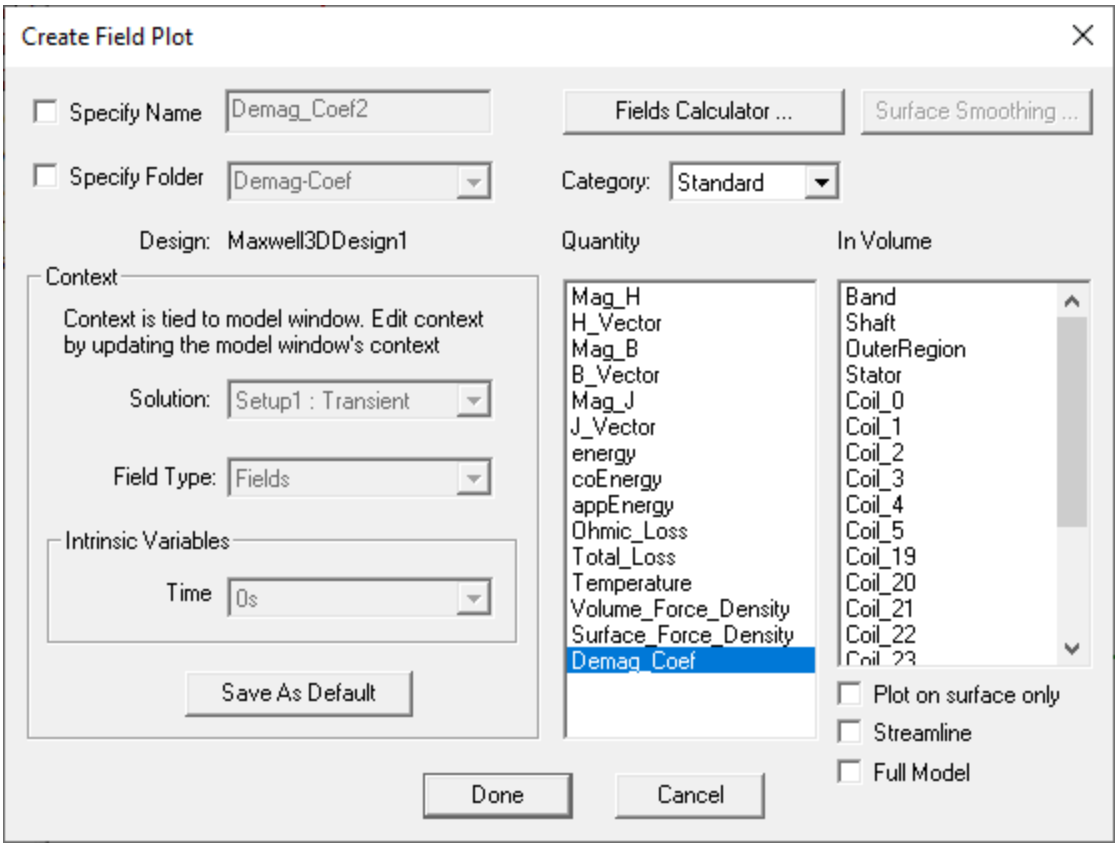
If it does not exist, create it.

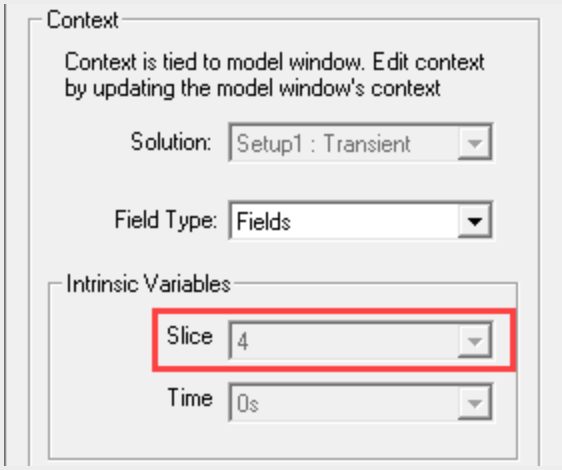
2. On the main menu bar, click **Maxwell3D>Fields>Fields** or **Maxwell2D>Fields>Fields**.
3. On the **Fields** sub-menu, click the field quantity you want to plot. Available selections depend on the solved solution. (You can also right-click on **Field Overlays** in the Project Manager and select **Fields>***field_quantity*).

   If you select a scalar field quantity, a scalar surface or volume plot is created. If you select a vector field quantity, a vector surface or volume plot is created. If you select a vector quantity, you will be able to specify a Streamline plot. If the quantity you want to plot is not listed, see *Calculating a Derived Field Quantity*.

   For projects with Temperature dependent materials, the **Maxwell>Fields>Fields>Other...** menus selections include **Temperature**.

   After you select the field quantity to plot, the **Create Field Plot** dialog box appears.



> **Note:** For Maxwell 2D transient solutions in which **Use Skew Model is enabled on the Model Settings tab**, an additional **Slice** field displays in the **Intrinsic Variables** panel.
>
> 
>

The **Specify Name** field shows a name based on the field quantity you selected, and the **Quantity** list shows the field quantity selected.

4. To specify a name for the plot other than the default, select **SpecifyName**, and then type a new name in the **Name** box.
5. To specify a folder other than the default in which to store the plot, select **Specify Folder**, and then click a folder in the pull-down list. Plot folders are listed under **Field Overlays** in the project tree. Plot folders let you group plots with the same quantity together. All field plots under the same folder share the same color key.

> **Note:** All plots (field overlays) in the same folder have the same scale settings. To plot the same field with a different scale, you can create or move the new plot to a separate folder. By default, current density plots are stored in a folder called **J**, but you can specify a different or new folder. Plots in different folders have a different plot keys.

6. Select the solution to plot from the **Solution** pull-down list.
7. Select the **Fields** type to plot in the **Field Type** pull-down list.
8. Under **Intrinsic Variables**, select the variable value(s) at which the field quantity is evaluated. Variables available vary depending on the solution type.

> **Note:** For Maxwell 2D transient solutions in which **Use Skew Model is enabled on the Model Settings tab**, a **Slice** field displays in the **Intrinsic Variables** panel showing the slice currently selected for viewing in the modeler window.

9. If desired, you can choose a different field quantity to plot from the **Quantity** list. For scalar quantities plotted on the surface of a geometry, the **Surface Smoothing** button is enabled.

   > **Note:** If you selected an *object*, you must check **Plot on surface only** to enable the **Surface Smoothing** button.

   For vector quantities, the **Surface Smoothing** button is disabled. For details on using the **Surface Smoothing** button, see Modifying Field Plots.

10. Select the volume or surface (region) in which to plot the field from the **In Volume** list.

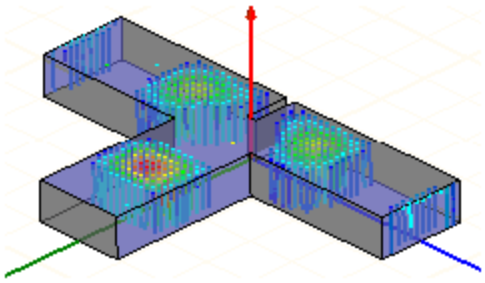
    This selection enables you to limit plots to the intersection of a volume with the selected object or objects. You can select and deselect any items in the **In Volume** list. You can mix model objects with non-model boxes. For example you might want to see a plot from part of two model objects by restricting the region to a non-model box overlapping those parts.

    > **Note:** Multiple selection should be used when there is a discontinuous field on a surface. If not, the field on both sides of the surface is plotted and each interferes with the other.

    > **Note:** Non-convex 3D non-model solids are not supported for Field Overlay Plots and Fields Calculator computations.

    > **Note:** Object lists containing non-model object(s) are supported for Field Overlay Plots but are not supported for Fields Calculator computations.

11. If you selected a vector quantity, you can use the check box to select **Streamline** plot. Streamlines are often used to indicate magnetic flux lines, etc. in plots.



Note	• Before creating the Streamline plot, select both starting and ending edges (in 2D) or both surfaces (in 3D). • In the **Create Field Plot** dialog box, select **In Volume:** region, which is the region in which the streamlines will appear. Streamlines are outside of objects that are excited (sources).

	• To show more streamlines after the plot is created, on the **Modify attributes...** dialog **Plots** tab, reduce the **Seeds density** value. If no streamlines appear, reduce this by a factor of 10 (or 100) because the default seeding was too large. Refer to Setting Field Plot Attributes for adjusting the Streamline display parameters, and Setting Fields Reporter Options for setting Streamline defaults. . For exporting Streamline plots, refer to Exporting Field Plots.

12. *Optionally*, you can select the **Plot on surface only** check box to obtain a plot around the surfaces of selected objects.
13. *Optionally*, for model objects in partial model designs, click the **Full Model** check box to calculate the fields on the full model based on the partial model solution.
14. *Optionally*, click **Fields Calculator** to open the fields calculator in which you can create named expressions, which can then be selected in the Create Field Plot dialog by choosing **Calculator** as the Category. The named expression(s) appear in the **Quantity** list.
15. Click **Done**.

    The field quantity is plotted on the surfaces or within the objects you selected. The plot uses the attributes specified in the **Plot Attributes** dialog box.

    The new plot appears in the view window. It is listed in the specified plot folder in the project tree. If you have created a field plot on a simulation in progress, the field plot is updated after the last adaptive solution.

    If you want to update the field overlay before then, to view progress in the solution, select the Field icon in the Project tree that contains the field plot of interest, right-click to display the short cut menu, and select **Update Plots**.

    To turn off the display of the plot, right click on the plot and select **Plot Visibility** from the short-cut menu. Unchecking **Plot Visibility** turns off the plot display.

Related Topics

Plotting Noise Vibration Overlays
Plotting Standard Field Quantities
Plotting Derived Field Quantities
Creating 2D Reports From Named Expressions
Calculating a Derived Field Quantity
Named Expressions
Creating Reports
Using the Fields Calculator
Setting a Plot's Visibility

Plotting Standard Field Quantities

Maxwell makes numerous quantities available for field plotting derived directly from the field solution. The quantities available depend upon the solution type (magnetostatic, eddy current, etc), on the design type (3D or 2D) and other parameters such as the use of lossy materials. The chart below lists possible solution quantities available when using **Maxwell3D>Fields>Fields**, or **Maxwell2D>Fields>Fields**.

Description	Quantity	2D	3D
Vector Potential	**Flux_Lines, A_Vector**	Magnetostatic, Eddy Current, Transient	
Magnetic Field Intensity	**Mag_H, ComplexMag_H, H_ Vector**	Magnetostatic, Eddy Current, Transient	Magnetostatic, Eddy Current, Transient
Magnetic Flux Density	**Mag_B, ComplexMag_B,B_ Vector**	Magnetostatic, Eddy Current, Transient	Magnetostatic, Eddy Current, Transient
Current Density	**JatPhase, Mag_J, ComplexMag_J, J_ Vector, Jphi, Jz**	Magnetostatic, Eddy Current, Transient, ACConduction, DCConduction	Magnetostatic, Eddy Current, Transient, DCConduction, Electric Transient, ACConduction
Direct Current Density	**Mag_Jc, ComplexMag_Jc, Jc_ Vector**		ACConduction
Electric Flux Density	**DatPhase, Mag_D, ComplexMag_D, D_ Vector**	Eddy Current, Electrostatic, ACConduction	Eddy Current, Electrostatic, DCConduction, Electric Transient, ACConduction
Electric Field Intensity	**Mag_E, ComplexMag_E, E_ Vector**	Electrostatic, ACConduction, DCConduction	Eddy Current, Electrostatic, DCConduction, Electric Transient
Surface Charge Density	**QSurf**		Electrostatic, ACConduction, DCConduction, Electric Transient
Surface Current Density (on an impedance boundary)	**Jsurf**		Eddy Current

Description	Quantity	2D	3D
Voltage	**Voltage**	Electrostatic, ACConduction, DCConduction	Electrostatic, DCConduction, Electric Transient, ACConduction, Transient A-Phi
Energy in the Field **Electric** **Magnetic**	**energy**	Magnetostatic, Eddy Current, Transient, Electrostatic	Magnetostatic, Eddy Current, Transient, Electrostatic, ACConduction, DCConduction, Electric Transient
Magnetic CoEnergy	**coEnergy**	Magnetostatic, Eddy Current, Transient	Magnetostatic, Transient
	appEnergy	Magnetostatic, Eddy Current, Transient	Magnetostatic, Transient
Ohmic Loss	**Ohmic_Loss**	Magnetostatic, Eddy Current, Transient, Electrostatic, ACConduction, DCConduction	Magnetostatic, Eddy Current, Transient, DCConduction, Electric Transient, ACConduction
Total Loss	**Total_Loss**	Transient (Magnetic)	
Hysteresis Loss	**Hysteresis_Loss**	Eddy Current	Eddy Current
Dielectric Loss	**Dielectric_Loss**	Eddy Current	Eddy Current ACConduction
Surface Loss Density	**SurfaceLossDensity**		Eddy Current
Edge Force Density	**edgeForceDensity**	Magnetostatic, Eddy Current (DC component only), Transient, Electrostatic	
Temperature	**Temp**	Magnetostatic, Eddy Current, Electrostatic	Magnetostatic, Eddy Current, Electrostatic
Surface Force Density	**surfaceForceDensity**	Magnetostatic, Eddy Current (DC component only), Transient, Electrostatic	Magnetostatic, Eddy Current (DC component only), Transient, Electrostatic, Electric Transient
Volume Force Density	**volumeForceDensity**		Magnetostatic, Eddy Current, Transient,

Description	Quantity	2D	3D
			Electrostatic, Electric Transient
Displacement	**Displacement_ Vector, Mag_ Displacement**	Magnetostatic, Eddy Current, Electrostatic	Magnetostatic, Eddy Current, Electrostatic
Sum of Ohmic, Hysteresis, and Dielectric Loss	**emloss**	Eddy Current	
Loss on Impedance boundary applied to a conductor	**edgeLossDensity**	Eddy Current	
Demagnetization coefficient	**Demag_coef**	Magnetostatic, Transient	Magnetostatic, Transient
Loss Density	**LossDensity**	Transient	Transient

Related Topics

Plotting Derived Field Quantities
Creating 2D Reports From Named Expressions
Calculating a Derived Field Quantity
Named Expressions
Creating Reports
Using the Fields Calculator

Force Density Quantities

Edge Force Density	The edge density is available only in Maxwell2D. It is available only on object outlines adjacent to air or vacuum space. It is defined as: $f = \overline{S} \cdot n$

	$$\overline{S} = \frac{1}{\mu_0}\begin{bmatrix} B_x^2 - B^2/2 & B_x B_y \\ B_x B_y & B_y^2 - B^2/2 \end{bmatrix} + \epsilon_0 \begin{bmatrix} E_x^2 - E^2/2 & E_x E_y \\ E_x E_y & E_y^2 - E^2/2 \end{bmatrix}$$ where **B** and **E** are the magnetic flux density and electric field intensity at the midpoint of the edge, respectively, and **n** is the outward normal direction of the edge. For electrostatic, AC, and DC solvers, only the electric field is included; while for magnetostatic, Eddy current, and magnetic transient solvers, only the magnetic field is considered. For AC solver and Eddy current solver, edge force density is the averaged (DC) force density.
Surface Force Density	In Maxwell2D, surface force density is defined as: $$\boldsymbol{f} = \frac{\sum_{i=0}^{i=N} \overline{S_i} \cdot \boldsymbol{n_i}\, l_i r_i}{area\ of\ element * r_c}$$ $$\overline{S_i} = \frac{1}{\mu}\begin{bmatrix} B_x^2 - B^2/2 & B_x B_y \\ B_x B_y & B_y^2 - B^2/2 \end{bmatrix} + \epsilon \begin{bmatrix} E_x^2 - E^2/2 & E_x E_y \\ E_x E_y & E_y^2 - E^2/2 \end{bmatrix}$$ where N is the number of sides of the element, l_i is the length of side i of the element. For the RZ case, r_i and r_c are the radius of side i and the radius of the element center, respectively. μ and ϵ are the permeability and permittivity in the ith neighboring element. In the XY plane, r_i and r_c are set to 1. In Maxwell3D, surface force density is only available on object faces adjacent to air or vacuum space. It is defined as: $$\boldsymbol{f} = \overline{S} \cdot \boldsymbol{n}$$ $$\overline{S} = \frac{1}{\mu_0}\begin{bmatrix} B_x^2 - B^2/2 & B_x B_y & B_x B_z \\ B_x B_y & B_y^2 - B^2/2 & B_z B_y \\ B_x B_z & B_z B_y & B_z^2 - B^2/2 \end{bmatrix} + \epsilon_0 \begin{bmatrix} E_x^2 - E^2/2 & E_x E_y & E_x E_z \\ E_x E_y & E_y^2 - E^2/2 & E_z E_y \\ E_x E_z & E_z E_y & E_z^2 - E^2/2 \end{bmatrix}$$

	where **B** and **E** are the magnetic flux density and electric field intensity at the center of the surface face, respectively, and **n** is the outward normal direction of the surface face. For electrostatic, AC, and DC solvers, only the electric field is included; while for magnetostatic, Eddy current, and magnetic transient solvers, only the magnetic field is considered. For AC solver and Eddy current solver, surface force density is the averaged (DC) force density.
Volume Force Density	Volume force density is only available in Maxwell3D. It is calculated as: $$\boldsymbol{f} = \frac{\sum_{i=0}^{i=N} \overline{\boldsymbol{S_i}} \cdot \boldsymbol{n_i} A_i}{Volume\ of\ element} \qquad (7)$$ $$\overline{S_i} = \frac{1}{\mu}\begin{bmatrix} B_x^2 - B^2/2 & B_x B_y & B_x B_z \\ B_x B_y & B_y^2 - B^2/2 & B_z B_y \\ B_x B_z & B_z B_y & B_z^2 - B^2/2 \end{bmatrix} + \epsilon\begin{bmatrix} E_x^2 - E^2/2 & E_x E_y & E_x E_z \\ E_x E_y & E_y^2 - E^2/2 & E_z E_y \\ E_x E_z & E_z E_y & E_z^2 - E^2/2 \end{bmatrix}$$ μ and ϵ are the permeability and permittivity in the ith neighboring element. A_i is area of side i of element. For electrostatic, AC, and DC solvers, only the electric field is included; while for magnetostatic, Eddy current, and magnetic transient solvers, only the magnetic field is considered. For AC solver and Eddy current solver, volume force density is the averaged (DC) force density. Lorentz force density is calculated instead of volume force density from Equation (7) in pure conductors containing source current in magnetostatic, Eddy current, and magnetic transient solvers: $$\boldsymbol{f} = \boldsymbol{J} \times \boldsymbol{B}$$

Loss Quantities

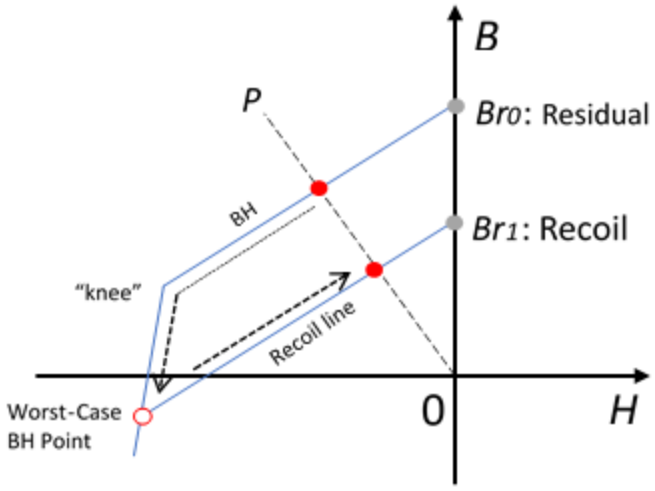
Ohmic loss	Ohmic loss is always associated with conduction current distribution in conductors which are not perfect. Thus the resistivity of conductors is responsible for the ohmic power loss when current flows in such conductors. It is also called the Joule-Lenz effect. There is always a heating effect due to the ohmic loss, often called Joule heating.

Total loss	Total loss is associated with loss density fields in 2D and 3D magnetic transient solutions only. Total loss uses the total integrated loss output from the solver and averages it, using each solver timestep between the last two consecutive save-field times. It is useful for passing loss density information from Maxwell to the Ansys thermal solvers with fewer save-field times. For Litz Wire only, total loss also includes StrandedLossAC. Unlike most other loss quantities which are instantaneous values, total loss is a running average over time, i.e. at time T1 it is calculated with fields from T1 to T2.
Hysteresis loss	Hysteresis loss is associated with loss density fields in 2D and 3D eddy current solutions only. Hysteresis loss is short for magnetic hysteresis loss and represents power loss in some magnetic materials (electric steels or ferrites) in alternating (sinusoidal) magnetic fields. This loss is due to a phenomenon called "magnetic viscosity" which causes the B and H fields to have a phase shift between them. In the B-H plane, for linear materials, the relationship between these two fields describes an ellipse. The hysteresis loss is proportional to the area of the ellipse.
Dielectric loss	Dielectric loss is associated with loss density fields in 2D and 3D eddy current solutions only. Dielectric loss or electric hysteresis loss represents power loss in ferroelectric materials (such as barium titanate, lead zirconate titanate) in alternating (sinusoidal) electric fields. This loss is due to a phenomenon called "electric viscosity" which causes the D and E fields to have a phase shift between them. In the D-E plane, for linear materials, the relationship between these two fields describes an ellipse. The dielectric loss is proportional to the area of the ellipse.
SurfaceLoss Density	The surface loss density is associated with the induced eddy current on the impedance boundary.
Emloss	In 2D and 3D eddy current solvers only: Emloss combines the ohmic loss, dielectric loss, hysteresis loss, and StrandedLossAC (for Litz Wire only).
Edgeloss density	Edge loss density represents losses associated with impedance boundary applied on conductive surfaces on the boundary of the solution domain or on surface of excluded objects. It assumes an exponential spatial decay (toward the interior of the respective material) of induced current magnitudes and of associated losses. Integrating the loss along the edge produces the entire loss associated with the "edge".
Core loss	The core loss combines eddy current losses and hysteresis losses for a transient solution type. It is a post-processing calculation, based on already calculated transient magnetic field quantities. It is applicable for the evaluation of core losses in steel laminations (frequently used in applications such as electric machines, transformers) or in power ferrites.
Solid Loss	The solid loss represents the resistive loss in a 2D or 3D volume and is calculated by:

	$$SolidLoss = \frac{1}{\sigma}\int_{vol} J^2 \cdot dV$$ For **2DXY** designs, the length is given by **Set Model Depth**, while for **RZ** designs the volume is based upon the rotation of the cross-section around the symmetry axis.
Stranded Loss (3 types)	There are three types of stranded loss quantities: **StrandedLoss**, **StrandedLossAC**, and **StrandedLossR**, which represent the loss in a 2D or 3D winding utilizing multi-turn coil terminals. • **StrandedLoss** is dependent on the object's material conductivity of a stranded source. It represents the loss due to the DC resistance only and uses the cross-sectional area of the object that is assigned with the excitation. Where ***J*** is the current density, it is calculated by: $$StrandedLoss = \frac{1}{\sigma}\int_{vol} J^2 \cdot dV$$ ◦ When Litz wire *is not defined*, StrandedLoss is computed by assuming fill factor = 100% (the ratio of net conductor area to coil terminal area). ◦ When Litz wire *is defined*, StrandedLoss is computed by using the actual fill factor based on the specified Litz stranding and represents the DC loss only. ◦ StrandedLoss does not include the losses of end-coil resistance in 2D. ◦ StrandedLoss can be used to scale the loss density to a coupled thermal analysis. • **StrandedLossAC** is reported when using Litz wire model, and is dependent on the object's material conductivity of a stranded source. It represents the loss due to the DC resistance as well as AC eddy current losses due to proximity effects and skin effects. ◦ StrandedLossAC = StrandedLoss + AC eddy current losses due to proximity effects and skin effects. ◦ StrandedLossAC does not include the losses of end-coil resistance in 2D.

	◦ StrandedLossAC can be used to transfer loss density to a coupled thermal analysis. • **StrandedLossR** represents the loss based on I^2 times the resistance **R**, by using the resistance value entered in the Winding properties for a voltage type winding only (Reported for a stranded voltage source not in an external circuit).
LossDensity	For Time Averaged Fields (for transient 2D and 3D solutions), the loss density is the time averaged loss density between a start time and end time, i.e., the accumulated loss density from start time to end time divided by the time duration.

Demagnetization Coefficient Plots



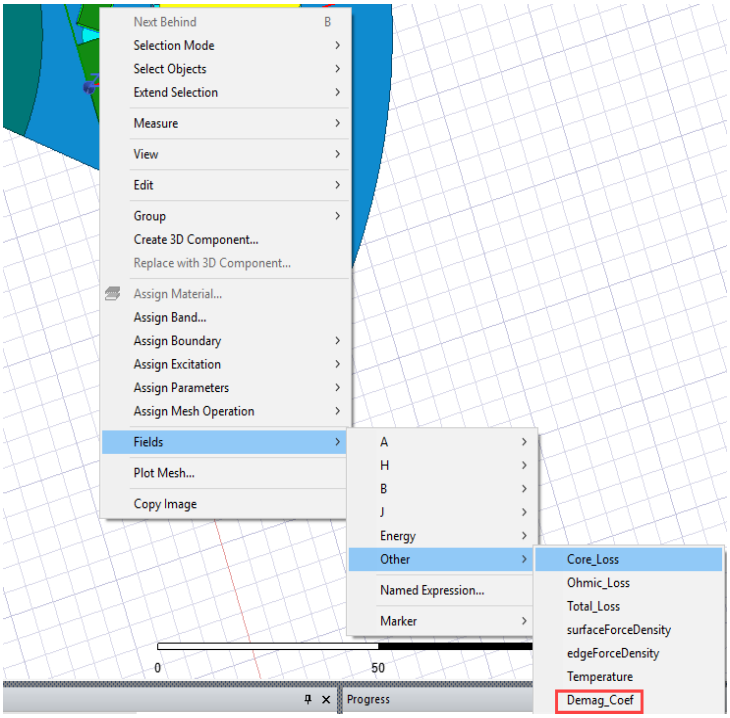
Demagnetization Coefficient: Demag_coef[%] = Br1/Br0*100%

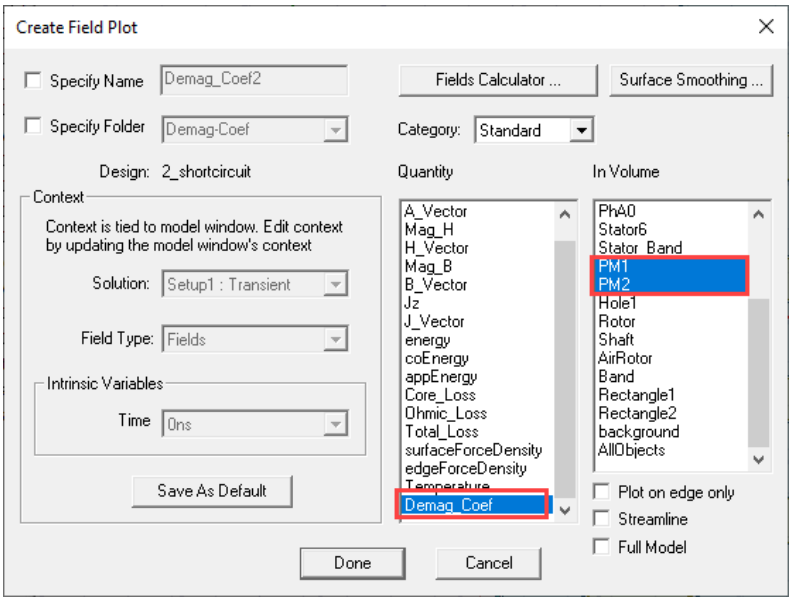
When Demag_Coef = 1, the magnet behaves without any demagnetization at that location;

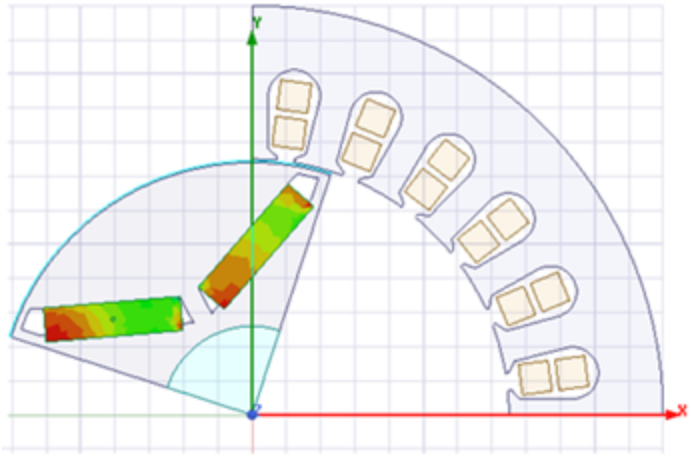
When Demag_Coef = 0, the magnet is fully demagnetized at that location.

Demag_Coef plot over the geometry of nonlinear permanent magnets:

Demag_Coef is a scalar value between 0 and 1, and is displayed as a color shaded plot inside nonlinear permanent magnets. **Demag_Coef** is applicable to 2D/3D transient solutions, and 2D/3D magnetostatic solutions - assign to objects by right-clicking them and selecting **Fields>Other>Demag_Coef**. Solvers provide the quantity at each node element-by-element to the post-processor for all nonlinear permanent magnetic objects at each time step with the saved field.

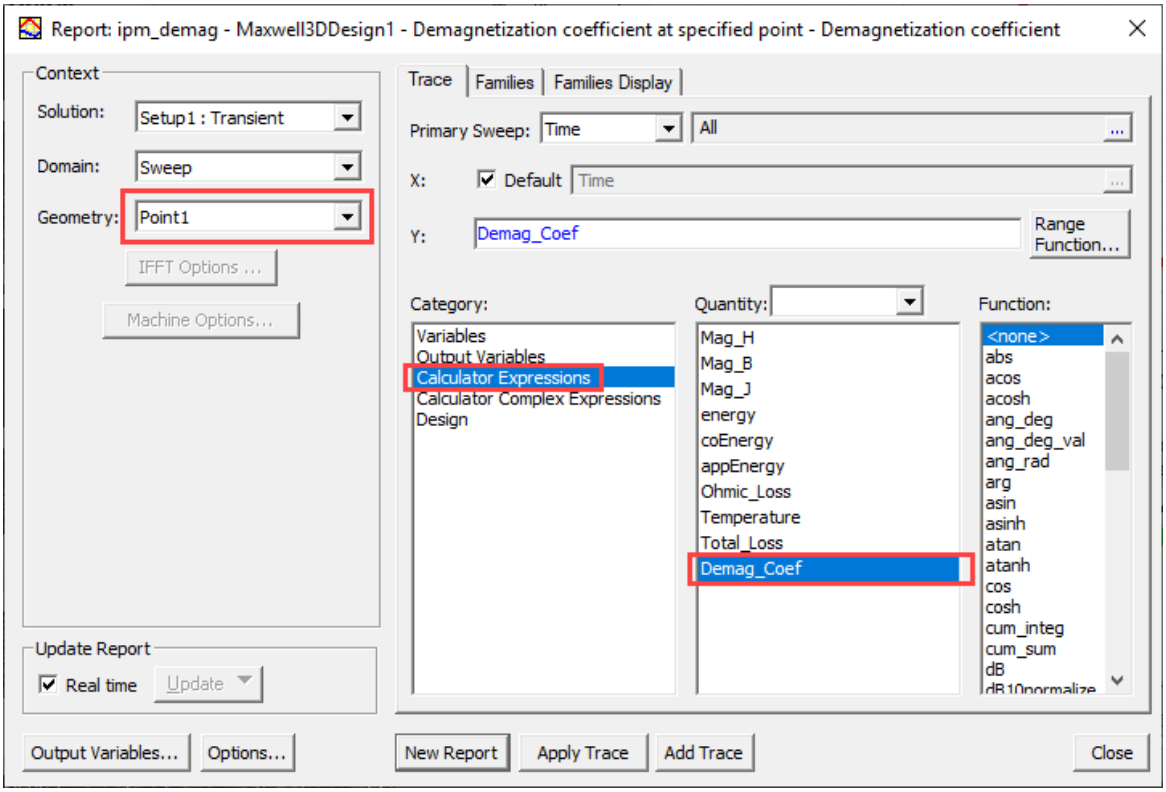


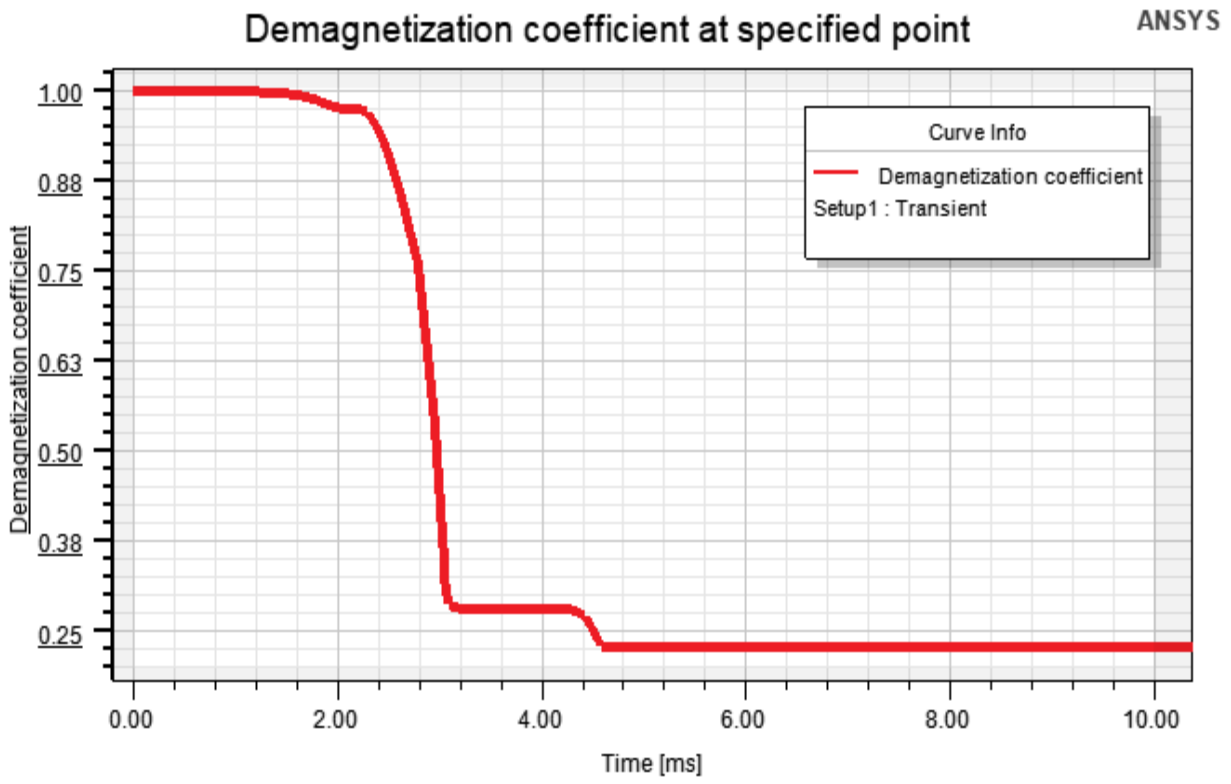




Demag_Coef curve as a function of time at specific locations:

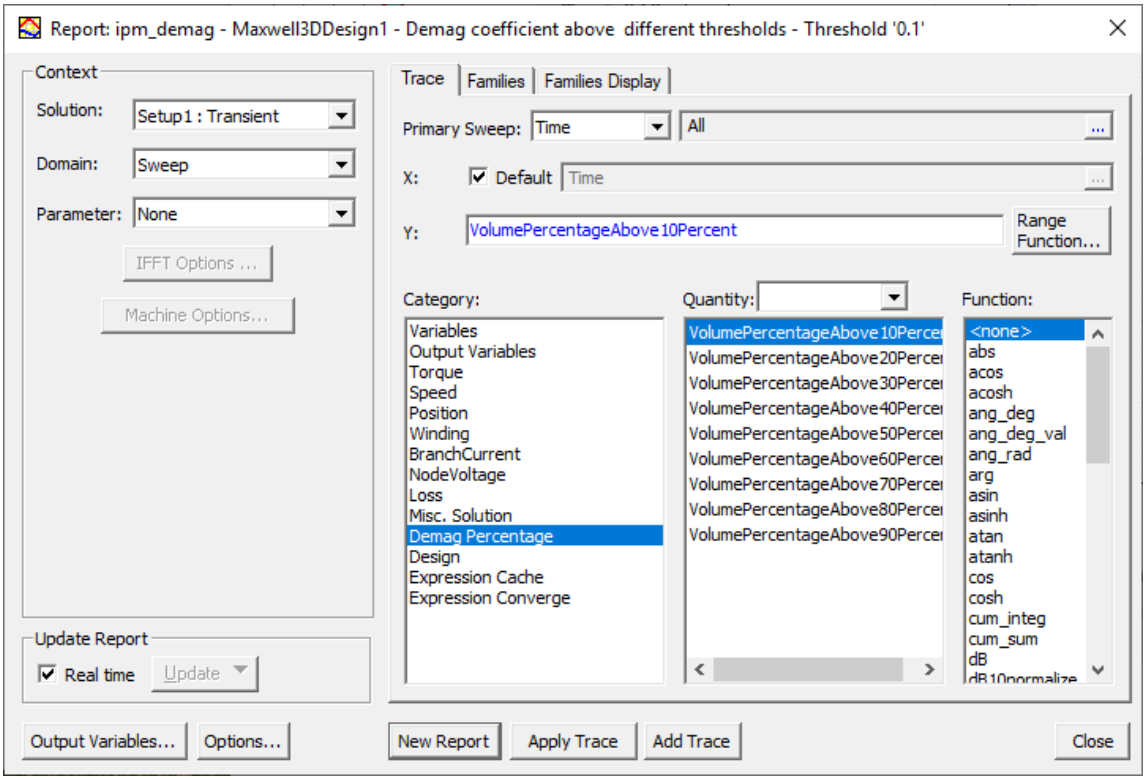
This is provided in terms of the expression cache at any specified point in the magnet. This plot is only applicable to 2D/3D transient solutions.



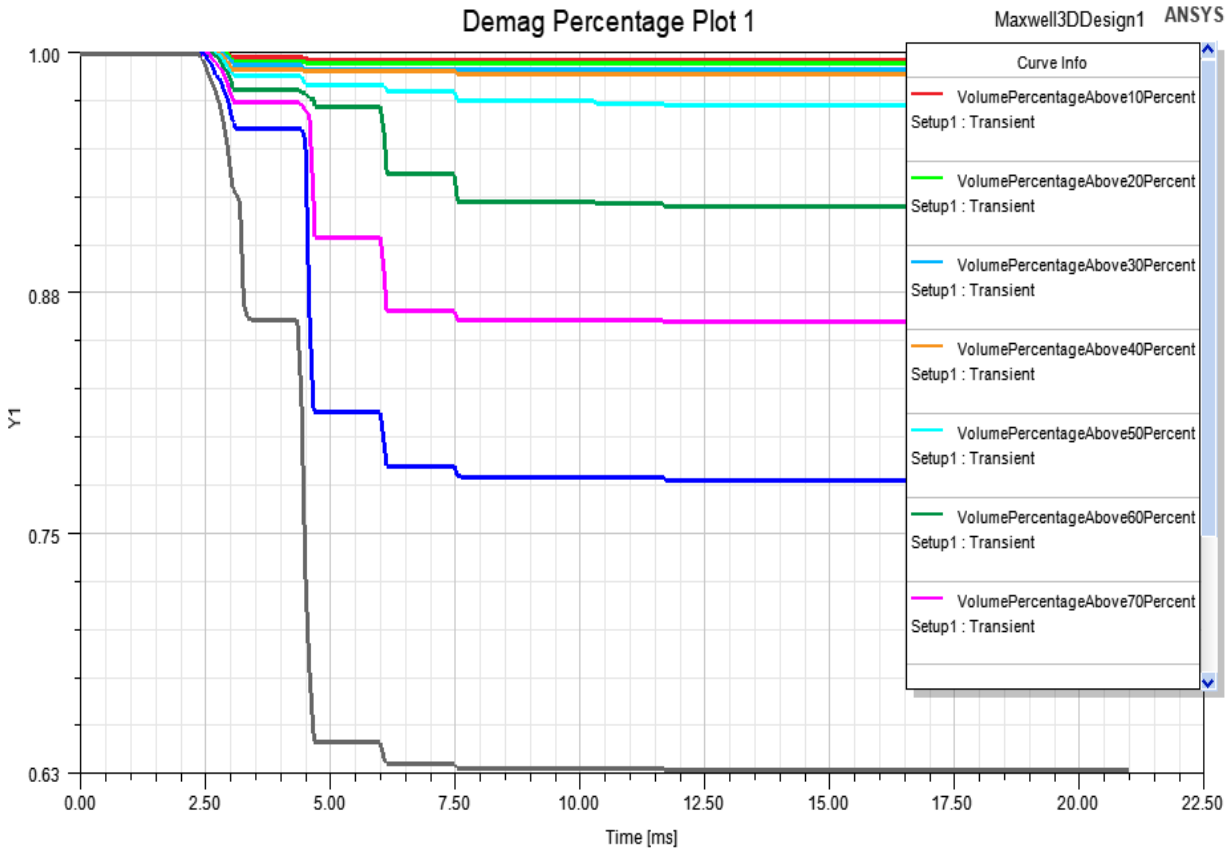


Demagnetization Percentage statistics curve as a function of time over all magnet volumes

For a specified Demag_Coef threshold, determine what the *percentage of total volume* for all magnets is, whose Demag_Coef is equal or greater than the specified Demag_Coef threshold. To determine this, the solver uses the Demag_Coef threshold values from 0.1 to 0.9, with 0.1 as the step interval, to compute the **Demag Percentage** statistics (**VolumePercentageAbove10Percent** through **VolumePercentageAbove90Percent**) at each time step.



As a result, the output is a series of nine curves from 0.1 to 0.9 similar to the example shown below. This plot is only applicable to 2D/3D transient solutions.



Plotting Time Averaged Fields Overlays

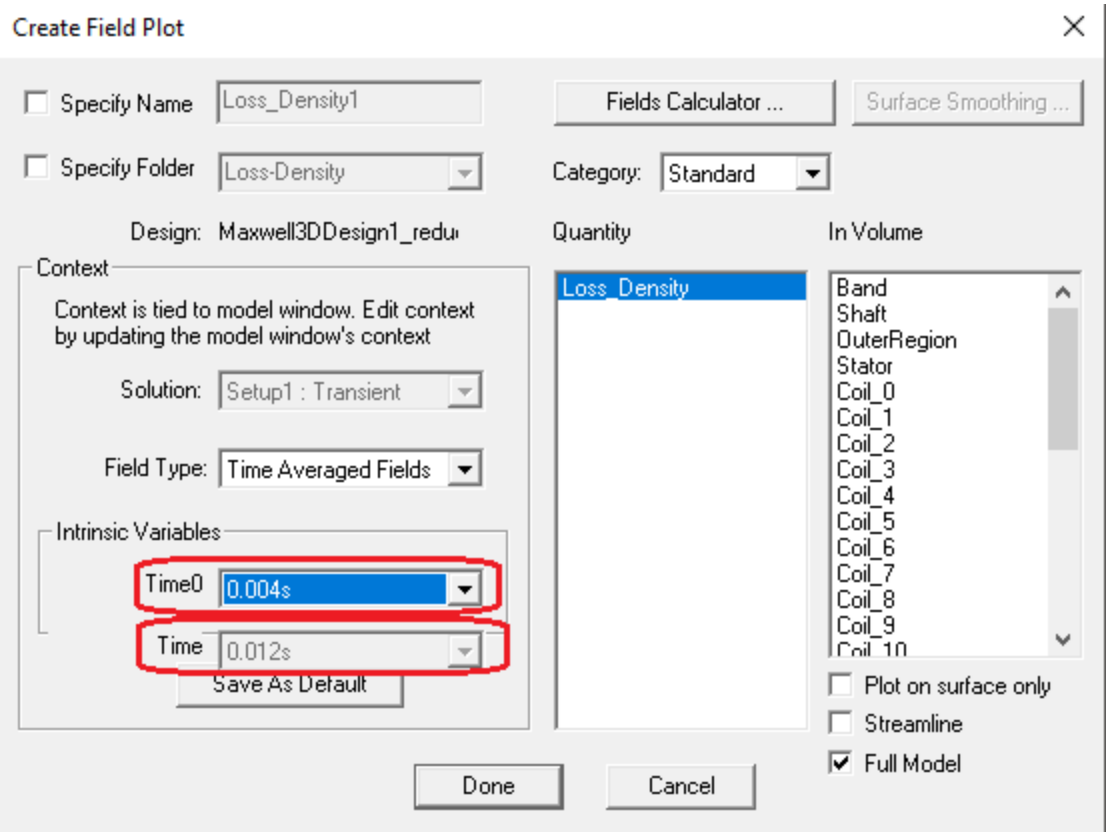
Time Averaged Field overlays are representations of time averaged field quantities, such as LossDensity, on surfaces or objects between specified start and end times divided by the time duration.

To plot a time averaged field quantity:

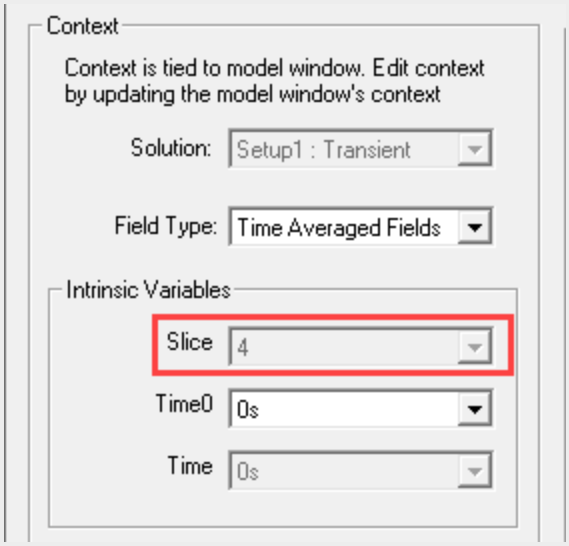
1. Select a point, line, surface, or object to create the plot on or within. You may also select a plane or object list in the **History Tree**.
2. On the main menu bar, click **Maxwell3D>Fields>Time Averaged Fields** or **Maxwell2D>Fields>Time Averaged Fields**.
3. On the **Time Averaged Fields** sub-menu, click the field quantity you want to plot. Available selections depend on the solved solution. (You can also right-click on **Field Overlays** in the Project Manager and select **Time Averaged Fields>***field_quantity*).

   If you select a scalar field quantity, a scalar surface or volume plot is created. If you select a vector field quantity, a vector surface or volume plot is created. If you select a vector quantity, you will be able to specify a Streamline plot. If the quantity you want to plot is not listed, see *Calculating a Derived Field Quantity*.

   After you select the field quantity to plot, the **Create Field Plot** dialog box appears.



Note: For Maxwell 2D transient solutions in which **Use Skew Model is enabled on the Model Settings tab**, an additional **Slice** field displays in the **Intrinsic**



Variables panel.

The **Specify Name** field shows a name based on the field quantity you selected, and the **Quantity** list shows the field quantity selected.

4. To specify a name for the plot other than the default, select **Specify Name**, and then type a new name in the box.
5. To specify a folder other than the default in which to store the plot, select **Specify Folder**, and then click a folder in the pull-down list. Plot folders are listed under **Field Overlays** in the project tree. Plot folders let you group plots with the same quantity together. All field plots under the same folder share the same color key.

Note	All plots (field overlays) in the same folder have the same scale settings. To plot the same field with a different scale, you can create or move the new plot to a separate folder. Plots in different folders have a different plot keys.

6. Select the solution to plot from the **Solution** pull-down list.
7. Select the fields type to plot in the **Field Type** to plot in the pull-down list. Choices are **Fields** or **Time Averaged Fields**.
8. Under **Intrinsic Variables**, select the variable value(s) at which the field quantity is evaluated. Variables available vary depending on the solution type.

> **Note:** For Maxwell 2D transient solutions in which **Use Skew Model is enabled on the Model Settings tab**, a **Slice** field displays in the **Intrinsic Variables** panel showing the slice currently selected for viewing in the modeler window.

9. If desired, you can choose a different field quantity to plot from the **Quantity** list.

Note	If you selected an *object*, you must check **Plot on surface only** to enable the **Surface Smoothing** button.

For vector quantities, the **Surface Smoothing** button is disabled. For details on using the **Surface Smoothing** button, see Modifying Field Plots.

10. Select the volume or surface (region) in which to plot the field from the **In Volume** list.

This selection enables you to limit plots to the intersection of a volume with the selected object or objects. You can select and deselect any items in the **In Volume** list. You can mix model objects with non-model boxes. For example you might want to see a plot from part of two model objects by restricting the region to a non-model box overlapping those parts.

Note	Multiple selection should be used when there is a discontinuous field on a surface. If not, the field on both sides of the surface is plotted and each interferes with the other.

11. If you selected a vector quantity, you can use the check box to select **Streamline** plot.

Note	• Before creating the Streamline plot, select both starting and ending edges (in 2D) or both surfaces (in 3D). • In the **Create Field Plot** dialog box, select **In Volume:** region, which is the region in which the streamlines will appear. Streamlines are outside of objects that are excited (sources).

	• To show more streamlines after the plot is created, on the **Modify attributes...** dialog **Plots** tab, reduce the **Seeds density** value. If no streamlines appear, reduce this by a factor of 10 (or 100) because the default seeding was too large. Refer to Setting Field Plot Attributes for adjusting the Streamline display parameters, and Setting Fields Reporter Options for setting Streamline defaults. . For exporting Streamline plots, refer to Exporting Field Plots.

12. *Optionally*, you can select the **Plot on surface only** check box to obtain a plot around the surfaces of selected objects.
13. *Optionally*, for model objects in partial model designs, click the **Full Model** check box to calculate the fields on the full model based on the partial model solution.
14. *Optionally*, click **Fields Calculator** to open the fields calculator in which you can create named expressions, which can then be selected in the Create Field Plot dialog by choosing **Calculator** as the Category. The named expression(s) appear in the **Quantity** list.
15. Click **Done**.

    The time averaged field quantity is plotted on the surfaces or within the objects you selected. The plot uses the attributes specified in the **Plot Attributes** dialog box.

    The new plot appears in the view window. It is listed in the specified plot folder in the project tree. If you have created a field plot on a simulation in progress, the field plot is updated after the last adaptive solution.

    If you want to update the field overlay before then, to view progress in the solution, select the Field icon in the Project tree that contains the field plot of interest, right-click to display the short cut menu, and select **Update Plots**.

    To turn off the display of the plot, right click on the plot and select **Plot Visibility** from the short-cut menu. Unchecking **Plot Visibility** turns off the plot display.

Related Topics

Plotting Noise Vibration Overlays
Plotting Standard Field Quantities
Plotting Derived Field Quantities
Creating 2D Reports From Named Expressions
Calculating a Derived Field Quantity
Named Expressions
Creating Reports
Using the Fields Calculator
Setting a Plot's Visibility

Plotting Noise Vibration Overlays *(Transient designs only)*

Noise vibration field overlays are representations of basic or derived noise vibration field quantities on surfaces or objects for the current design variation. You can set the design variation via the **Set Design Variation** dialog box. This dialog box is accessible from the **Solution Data** window via by clicking the ellipsis button on the right of the Design Variation field, and via the **Results>Apply Solved Variation** command.

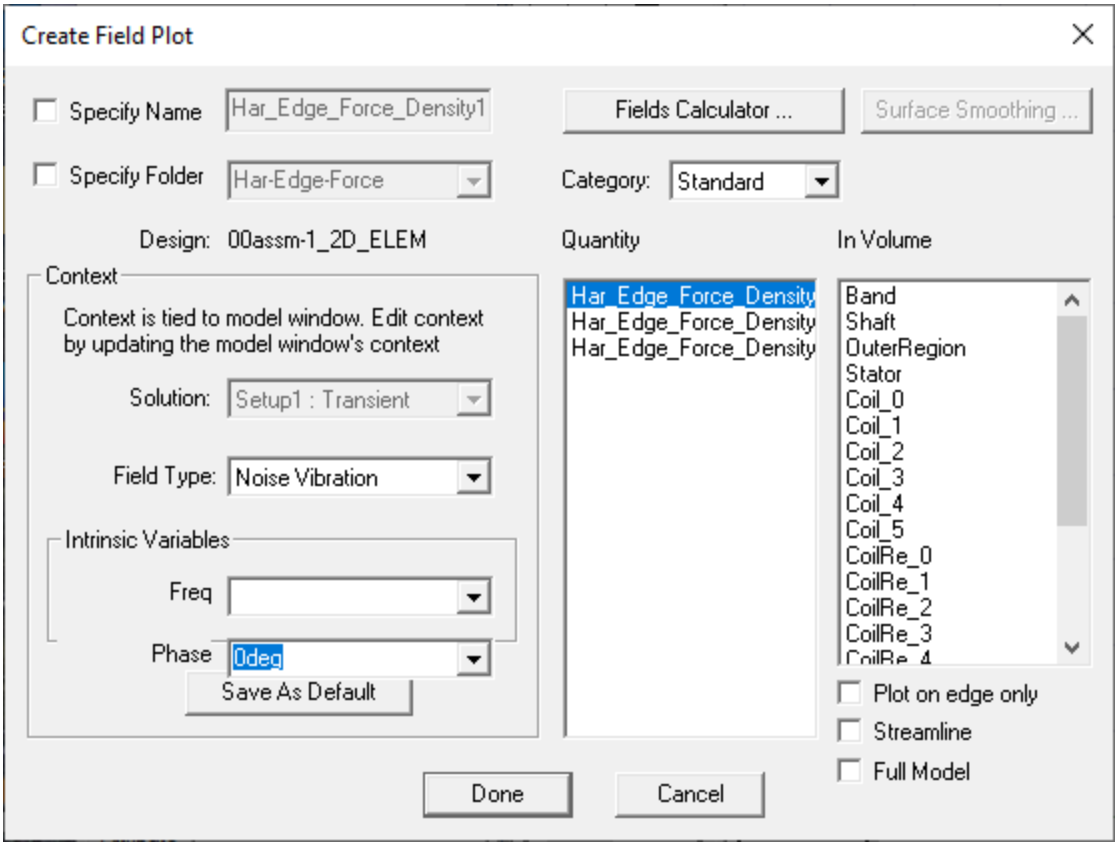
The **Noise Vibration** field type is available for Maxwell 2D and 3D Transient designs if element-based harmonic forces are set up. The **Noise Vibration** field type handles surface and volume force density in 3D, and surface and edge force density in 2D. **Frequency** and **Phase** intrinsic variables are used instead of **Time**. The **Noise Vibration** field overlays share the same Maxwell solution as the standard Fields overlays.

To plot a noise vibration field quantity:

1. Select a surface or volume for a 3D design; or an edge or surface for a 2D design, to create the plot on or within the selected objects.
2. On the main menu bar, click **Maxwell3D>Fields>Noise Vibration** or **Maxwell2D>Fields>Noise Vibration**.
3. On the **Noise Vibration** sub-menu, click the field quantity you want to plot. (You can also right-click on **Field Overlays** in the Project Manager and select **Fields>Noise Vibration>***field_quantity*).

   If you select a scalar field quantity, a scalar surface or volume plot is created. If you select a vector field quantity, a vector surface or volume plot is created. If you select a vector quantity, you will be able to specify a Streamline plot. If the quantity you want to plot is not listed, see *Calculating a Derived Field Quantity*.

   After you select the field quantity to plot, the **Create Field Plot** dialog box appears.



The **Specify Name** field shows a name based on the field quantity you selected, and the **Quantity** list shows the field quantity selected.

4. To specify a name for the plot other than the default, select **Specify Name**, and then type a new name in the box.
5. To specify a folder other than the default in which to store the plot, select **Specify Folder**, and then click a folder in the pull-down list. Plot folders are listed under **Field Overlays** in the project tree. Plot folders let you group plots with the same quantity together. All field plots under the same folder share the same color key.

Note	All plots (field overlays) in the same folder have the same scale settings. To plot the same field with a different scale, you can create or move the new plot to a separate folder. Plots in different folders have a different plot keys.

6. Select the solution to plot from the **Solution** pull-down list.
7. Select **Noise Vibration** in the **Field Type** pull-down list.
8. Under **Intrinsic Variables**, select the variable value(s) at which the field quantity is evaluated. For Noise Vibration, the variables available are **Freq**ency and **Phase**.
9. If desired, you can choose a different field quantity to plot from the **Quantity** list. For the **Har_Surf_Force_Density_Peak** quantity plotted on the surface of a geometry, the **Surface Smoothing** button is enabled. For the other two quantities, the **Surface**

Smoothing button is disabled. For details on using the **Surface Smoothing** button, see Modifying Field Plots.

10. Select the volume or surface (region) in which to plot the field from the **In Volume** list.

    This selection enables you to limit plots to the intersection of a volume with the selected object or objects. You can select and deselect any items in the **In Volume** list. You can mix model objects with non-model boxes. For example you might want to see a plot from part of two model objects by restricting the region to a non-model box overlapping those parts.

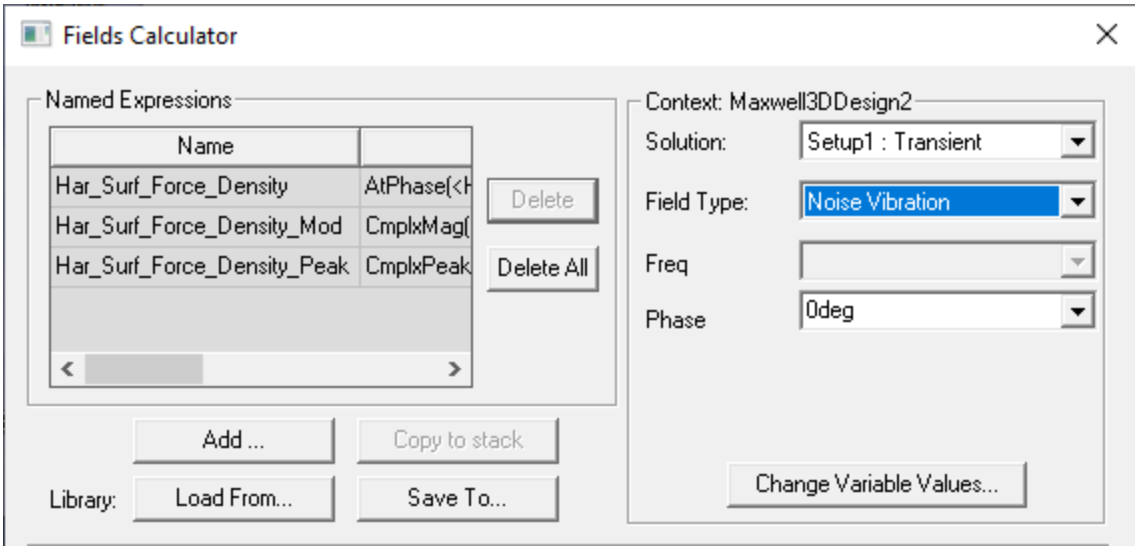
Note	Multiple selection should be used when there is a discontinuous field on a surface. If not, the field on both sides of the surface is plotted and each interferes with the other.

Note	• Non-convex 3D non-model solids are not supported for Field Overlay Plots and Fields Calculator computations. • Object lists containing non-model object(s) are supported for Field Overlay Plots but are not supported for Fields Calculator computations.

11. *Optionally*, You can use the check box to select **Streamline** plot.

Note	• Before creating the Streamline plot, select both starting and ending edges (in 2D) or both surfaces (in 3D). • In the **Create Field Plot** dialog box, select **In Volume:** region, which is the region in which the streamlines will appear. Streamlines are outside of objects that are excited (sources). • To show more streamlines after the plot is created, on the **Modify attributes...** dialog **Plots** tab, reduce the **Seeds density** value. If no streamlines appear, reduce this by a factor of 10 (or 100) because the default seeding was too large. Refer to Setting Field Plot Attributes for adjusting the Streamline display parameters, and Setting Fields Reporter Options for setting Streamline defaults. . For exporting Streamline plots, refer to Exporting Field Plots.

12. *Optionally*, for 3D designs you can select the **Plot on surface only** check box to obtain a plot around the surfaces of selected objects. Similarly, for 2D designs, you can select **Plot on edge only** to obtain a plot around the edges of selected objects.
13. *Optionally*, for model objects in partial model designs, click the **Full Model** check box to calculate the fields on the full model based on the partial model solution.
14. *Optionally*, click **Fields Calculator** to open the fields calculator in which you can create named expressions.



These can then be selected in the **Create Field Plot** dialog by choosing **Calculator** as the Category. The named expression(s) appear in the **Quantity** list.

15. Click **Done**.

    The field quantity is plotted on the surfaces or within the objects you selected. The plot uses the attributes specified in the **Plot Attributes** dialog box.

    The new plot appears in the view window. It is listed in the specified plot folder in the project tree. If you have created a field plot on a simulation in progress, the field plot is updated after the last adaptive solution.

    If you want to update the field overlay before then, to view progress in the solution, select the Field icon in the Project tree that contains the field plot of interest, right-click to display the short cut menu, and select **Update Plots**.

    To turn off the display of the plot, right click on the plot and select **Plot Visibility** from the short-cut menu. Unchecking **Plot Visibility** turns off the plot display.

Related Topics

Plotting Standard Fields Overlays

Calculating a Derived Field Quantity

Named Expressions

Creating Reports

Using the Fields Calculator

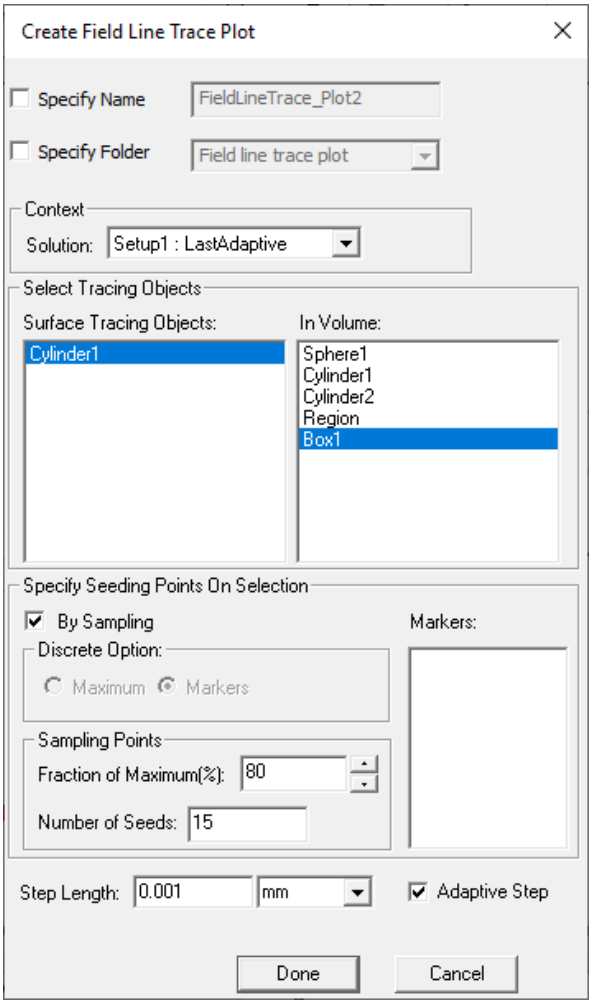
Setting a Plot's Visibility

Plotting Field Line Trace Overlays

For Maxwell electrostatic solutions, Field Line Trace overlays allow you to plot E field lines in 3D space and on insulator surfaces. The E field lines allow users to calculate the inception voltages for breakdown analyses.

To plot a field line trace overlay:

1. Select a surface from the geometry model or from a scalar surface plot.
2. Launch the **Create Field Line Trace Plot** dialog from the main **Maxwell>Fields>Field>Field Line Trace** menu, from the modeler context menu, or from the **Fields Overlay** context menu in the project tree.



3. To specify a name for the plot other than the default, select **Specify Name**, and then type a new name in the box. The **Specify Name** field shows a default name based on the field quantity you selected,
4. To specify a folder other than the default in which to store the plot, select **Specify Folder**, and then click a folder in the pull-down list. Plot folders are listed under **Field Overlays** in

the project tree. Plot folders let you group plots with the same quantity together. All field plots under the same folder share the same color key.

Note	All plots (field overlays) in the same folder have the same scale settings. To plot the same field with a different scale, you can create or move the new plot to a separate folder. Plots in different folders have a different plot keys.

5. Select the desired **Surface Tracing Objects** and **In Volume** from the respective lists. You can use the **Ctrl** and **Shift** keys to select multiple items.
6. The **Specify Seeding Points on Selection** panel allows you to manage the seeding points that will be used when calculating the trace lines:
    - If **By Sampling** is not selected, you can select either the **Maximum**, or **Markers** if markers for an E Field overlay plot have been set.
    - If **By Sampling** is selected, **Maximum** and **Markers** choices are disabled, and you can set a **Fraction of Maximum (%)** percentage value and the **Number of Seeds** to be used when calculating trace lines.
7. Enter the **Step Length** and unit. Step length controls the spacing between evaluation points along the E field direction. If **Adaptive Step** is checked, the step length will be inversely proportional to the E field strength.
8. Click **Done** to close the dialog box.

You can now analyze the design to generate the field line trace plot. After the field line trace plot has been generated, you can invoke the **Inception Voltage Evaluation** dialog box, which allows you to calculate the breakdown voltage - the voltage at which corona discharge will occur for the selected seed position and gas type.

Related Topics

Plotting Noise Vibration Overlays

Plotting Standard Field Quantities

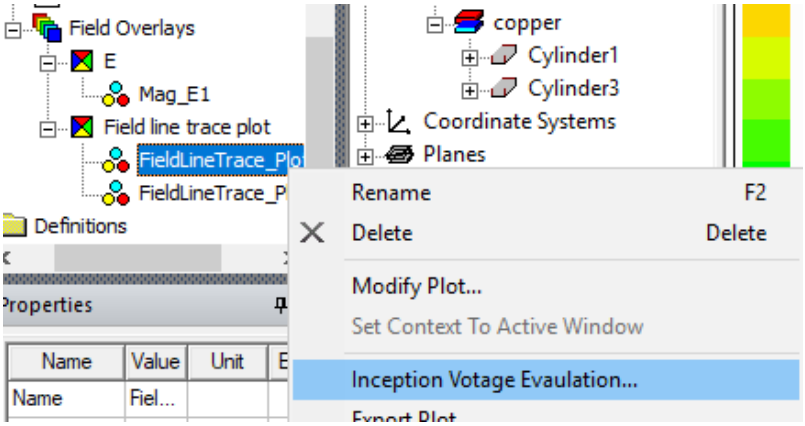
Inception Voltage Evaluation

Inception Voltage Evaluation

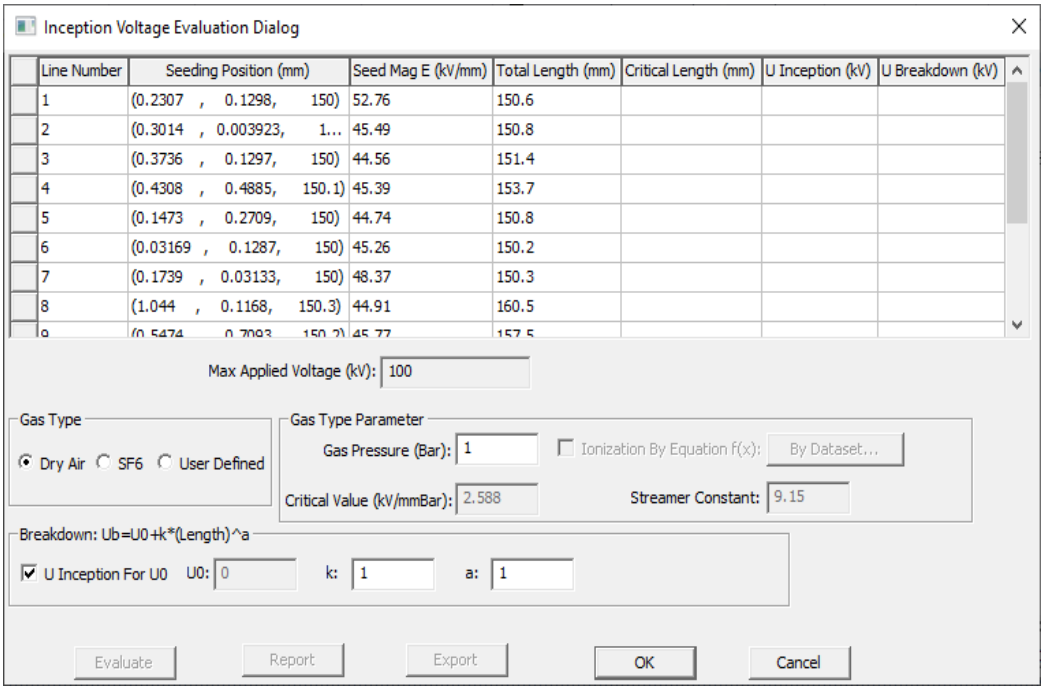
For Maxwell electrostatic solutions, Field Line Trace overlays allow you to plot E field lines in 3D space and on insulator surfaces. After a field line trace plot has been generated, you can invoke the **Inception Voltage Evaluation** dialog box, which allows you to calculate the breakdown voltage – the voltage at which a corona discharge will occur for the selected seed positions and gas type.

To evaluate inception voltages:

1. For any solved field line trace plot, launch the **Inception Voltage Evaluation Dialog** from the **Fields Overlay** context menu in the project tree.



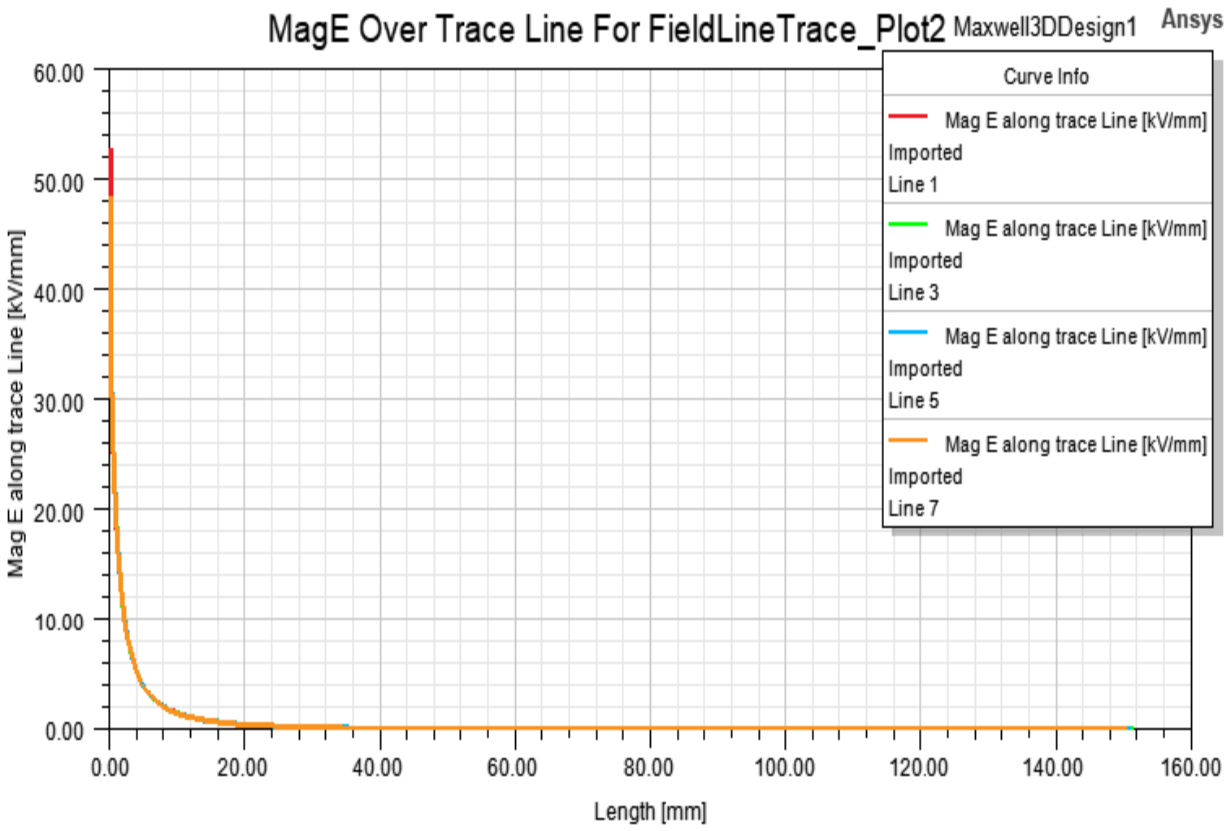
The **Inception Voltage Evaluation Dialog** opens.



2. The **Gas Type** panel allows you to choose either **Dry Air**, **SF6** (sulfur hexaflouride), or a **User Defined** gas. The selection controls which gas type parameters can be entered.
3. In the **Gas Type Parameter** panel:
    - For **Dry Air** and **SF6**, only the **Gas Pressure** value (in **Bar** units) can be adjusted. The **Critical Value** and **Streamer Constant** parameters are set to predetermined values.
    - For **User Defined** gas types, in addition to **Gas Pressure**, you can also set values for the **Critical Value** and **Streamer Constant** parameters. Additionally, you can click the **By Dataset** button to open an **Edit Dataset** dialog in which you can enter data

points that define the **Ionization Coefficient Curve** for the gas type you are using. If you check **Ionization By Equation f(x)**, you can specify a function for Ionization (for example, 16.8*x –81.0). Typically, the **User Defined** values would come either from experimental test results or from a manufacturer.

4. In the **Breakdown** panel with **U Inception for U0** unchecked, you can enter the **U0**, **k**, and **a** parameters for the breakdown equation where the **Length** is the total length of the concerned trace line. When **U Inception for U0** is checked, **U0** is the evaluated inception value.
5. Select the row(s) for which you wish to evaluate **Critical Length (mm)** and **U Inception (kV)**, and **U Breakdown (kV)** values. You can use the **Ctrl** and **Shift** keys to select multiple rows. You can select all rows by double clicking any cell. You can sort by column by clicking the column header, click it again to reverse the order.
6. Click **Evaluate** to have Maxwell generate the **Critical Length (mm)**, **U Inception (kV)**, and **U Breakdown (kV)** values for the selected rows. This data can be useful when choosing the insulation for a design.
7. You can select one or more rows from the Inception table to create a XY plot by clicking the “Report” button. A curve will be created for each selected trace line. It will be a curve of MagE of each point on the trace line vs the segment length of the point to the starting point of the trace line. For example:



8. When **U Inception (kV)** and **U Breakdown (kV)** are computed for any trace lines, the **Export** button is enabled. You can select any rows to export or export all computed rows if there is no selection. (However, if any of the selected lines don't have computed inception/breakdown data, the "Export" button will be disabled.) The data is exported as a text file.

   A sample exported text file (*.txt) is shown below, which includes the *Field Line Plot Inception Settings* that are used to compute the inception and breakdown voltages. It also includes the maximum applied voltage for the design and the inception table for the exported lines.

```
# Ansys ElectronicsDesktop 2022.1.0
# Field line plot: FieldLineTrace_Plot1, Inception data file, version 1.0

Field Line Plot Inception Settings:
Gas Type: Dry Air
Gas Pressure: 1.0000000000000000e+00
Use Inception: true
Potential U0: 0.0000000000000000e+00
Potential K: 1.0000000000000000e+00
Potential A: 1.0000000000000000e+00

Max applied voltage: 1.0000000000000000e+02(kV)
Line ID, Table Index, Seed Position(mm), Seed Mag E(kV/mm), Total Length(mm), Critical Length(mm), U Inception(kV), U Breakdown(kV)
0, 1, (2.3070381908095705e-01, 1.2979182877146406e-01, 1.5001759493820100e+02), 5.2762025637338382e+01, 1.5057577619847515e+02, 1.1716558116651508e+00, 1.4075577105973363e+01, 1.6465135330444852e+02
2, 3, (3.7362326532583046e-01, 1.2973613679416152e-01, 1.5003999575422500e+02), 4.4555986426930822e+01, 1.5138148173590182e+02, 1.2050596224905987e+00, 1.4098477232850046e+01, 1.6547995896875187e+02
3, 4, (4.3076251393666537e-01, 4.8847738196739027e-01, 1.5010958571062298e+02), 4.5386888094631729e+01, 1.5371815568704838e+02, 1.2026495369371775e+00, 1.4128121527752524e+01, 1.6784627721480089e+02
4, 5, (1.4733129741472195e-01, 2.7087188864013190e-01, 1.5002441955487799e+02), 4.4740231766632597e+01, 1.5082887732437518e+02, 1.1796928016782176e+00, 1.4082413189396163e+01, 1.6491129051377135e+02
```

Related Topics

Plotting Field Line Trace Overlays

Named Expressions

Named expressions can contain any combination of scalar, vector, or geometry functions. When you plot a field overlay or create a report based on a named expression, only corresponding quantities are available.

For example, if you select a point for the **Geometry** and a **Category** that is a named expression based on the point (or other scalar, non-3D value), then the **Quantity** list only lists expressions that returns a single scalar value. If you have added named expressions that take the magnitude of the ScalarX(), ScalarY(), or ScalarZ() of a vector point value, then you can create output variables for those expressions only in this case.

Named expressions can be plotted in the following three ways:

To create a field overlay of a named expression, the expression must be a real value (scalar or vector) that has values everywhere in space (or at least on every point in the geometry you plan to use for the field overlay).

To create a 2D report from a named expression that evaluates in the Fields Calculator as a single scalar value, the expression must result in a single-valued, real, scalar value. A single-valued item could be the value on a single point in space or the result of a function that returns a single value (such as an integration, max/value, min/value, or other function).

To create a 2D report from a named expression that is evaluated (in the Fields Calculator) along a polyline, the expression must be a real scalar that has values everywhere in space (or at least everywhere on the line object you want to use to sample the values).

Named expressions can be created in the Fields Calculator.

Plotting Derived Field Quantities

Derived field quantities are field quantity representations that have been deduced from the original field solution using the Fields Calculator.

1. Select a point, line, surface, or object to create the plot on or within. You may also select a plane or object list in the **History Tree**.

Note	For 2D Designs, a plane selection must be consistent with the drawing plane or an error will result.

   If it does not exist, create it.
2. Click **Maxwell3D or Maxwell2D and select Fields>Fields>Named Expression**.

   The **Selecting calculated expression** dialog box appears.
3. Select the derived quantity you want to plot, and then click **OK**.

   The **Create Field Plot** dialog box appears.
4. To specify a name for the plot other than the default, select **SpecifyName**, and then type a new name in the **Name** box.
5. To specify a folder other than the default in which to store the plot, select **Specify Folder**, and then click a folder in the pull-down list. Plot folders are listed under **Field Overlays** in the project tree.
6. Select the solution to plot from the **Solution** pull-down list.

7. Select the derived field quantity to plot from the **Quantity** list.

Note	If you select a point for the **Geometry** and a **Category** that is a named expression based on the point (or other scalar, non-3D value), then the **Quantity** list only lists expressions that returns a single scalar value. If you have added named expressions that take the magnitude of the ScalarX(), ScalarY(), or ScalarZ() of a vector point value, then you can create output variables for those expressions only in this case.

Note	If you select a polyline for the **Geometry** and a **Category** that is a named expression based on the line, then the **Quantity** list only lists corresponding expressions (i.e., will not list scalar values when a line is selected as the geometry).

8. Select the volume or surface (region) in which to plot the field from the **In Volume** list.

   This selection enables you to limit plots to the intersection of a volume and the selected object.

9. Click **Done**.

   The derived field quantity you created in the Fields Calculator is plotted on the surfaces or objects you selected. The new plot is listed in the project tree under **Field Overlays**.

Creating 2D Reports From Named Expressions

You can create a 2D report from named expressions that evaluate as a single scalar value or that evaluate along polylines.

To create a report from a named expression evaluating as a single scalar value:

1. Click **Maxwell2D or Maxwell3D>Results>Create Report**.

   The **Create Report** dialog box appears.

2. In the **Target Design** pull-down list, click the design containing the solution data you want to plot.
3. Select **Fields** from the **Report Type** pull-down list.
4. In the **Display Type** pull-down list, select the type of report you want to create.
5. Click **OK**.

   The **Traces** dialog box appears.

6. In the **Solution** pull-down list, click the solution containing the data you want to plot.
7. Add one or more traces to include in the report.

Note	If you select a point for the **Geometry** and a **Category** that is a named expression based on the point (or other scalar, non-3D value), then the **Quantity** list only lists expressions that returns a single scalar value. If you have added named expressions that take the magnitude of the ScalarX(), ScalarY(), or ScalarZ() of a vector point value, then you can create output variables for those expressions only in this case.

8. Click **Done**.

The report appears in the view window and is listed in the project tree. The default name based on the Report <type> you selected, (for example, Modal Solution Data Plot *n* or Far Fields Plot *n*). You can edit the plot names in the project tree and the plot header text in the report synchronizes. Once you have created a report, additional options become available on the **Results** submenu.

To create a report from a named expression evaluating along a polyline:

1. Click **Maxwell3D, Maxwell2D or RMxprt and select Results>Create Report**.

   The **Create Report** dialog box appears.
2. In the **Target Design** pull-down list, click the design containing the solution data you want to plot.
3. Select **Fields** from the **Report Type** pull-down list.
4. In the **Display Type** pull-down list, select the type of report you want to create.
5. Click **OK**.

   The **Traces** dialog box appears.
6. In the **Solution** pull-down list, click the solution containing the data you want to plot.
7. Select the geometry you want to plot from the **Geometry** pull-down list.
8. Add one or more traces to include in the report.

Note	If you select a polyline for the **Geometry** and a **Category** that is a named expression based on the line, then the **Quantity** list only lists corresponding expressions (i.e., will not list scalar values when a line is selected as the geometry).
9. Click **Done**.

   The report appears in the view window and is listed in the project tree. The default name is based on the Report Category you selected, (for example, Force Plot *n* or Output Variables Plot *n*). You can edit the plot names in the project tree and the plot header text in the report synchronizes. Once you have created a report, additional options become available on the **Results** submenu.

Creating Scalar Field Plots

A scalar plot uses shaded colors or contoured lines to illustrate the magnitude of field quantities on surfaces or volumes.

1. Do one of the following:
   a. To create a scalar surface plot, select the faces on which you want to plot the fields.
   b. To create a scalar volume plot, select the objects within which you want to plot the fields.
2. Click **Maxwell3D or Maxwell2D and then select Fields>Fields**.
3. On the **Fields** submenu, click the scalar field quantity you want to plot.

   The **Create Field Plot** dialog box appears.
4. Follow the procedure for plotting field overlays.

The new plot is listed in the specified plot folder in the project tree.

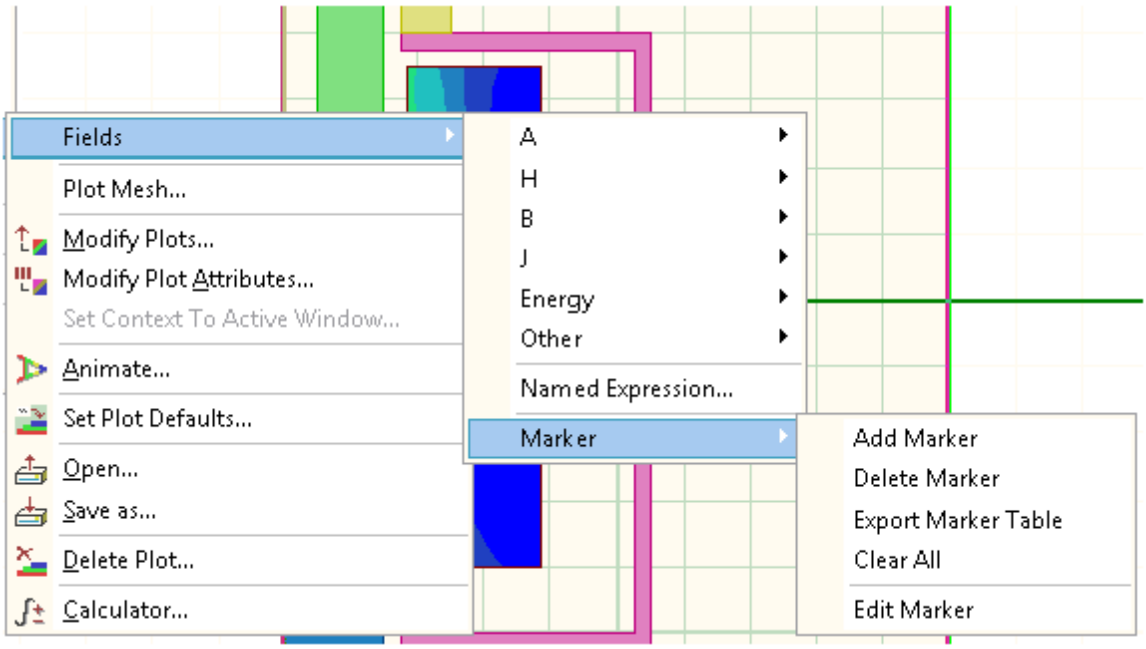
Related Topics

Modifying Field Plot Attributes

Working with Scalar Field Plot Markers

The field overlay plot marker feature enables you to create a marker at selected points in the scalar field overlay plot geometry, and to obtain the field value at that point. The fields Marker sub-menu enables you to:

- Add Marker
- Delete Marker
- Export Marker Table
- Clear All markers
- Edit Marker



Adding a Field Plot Marker

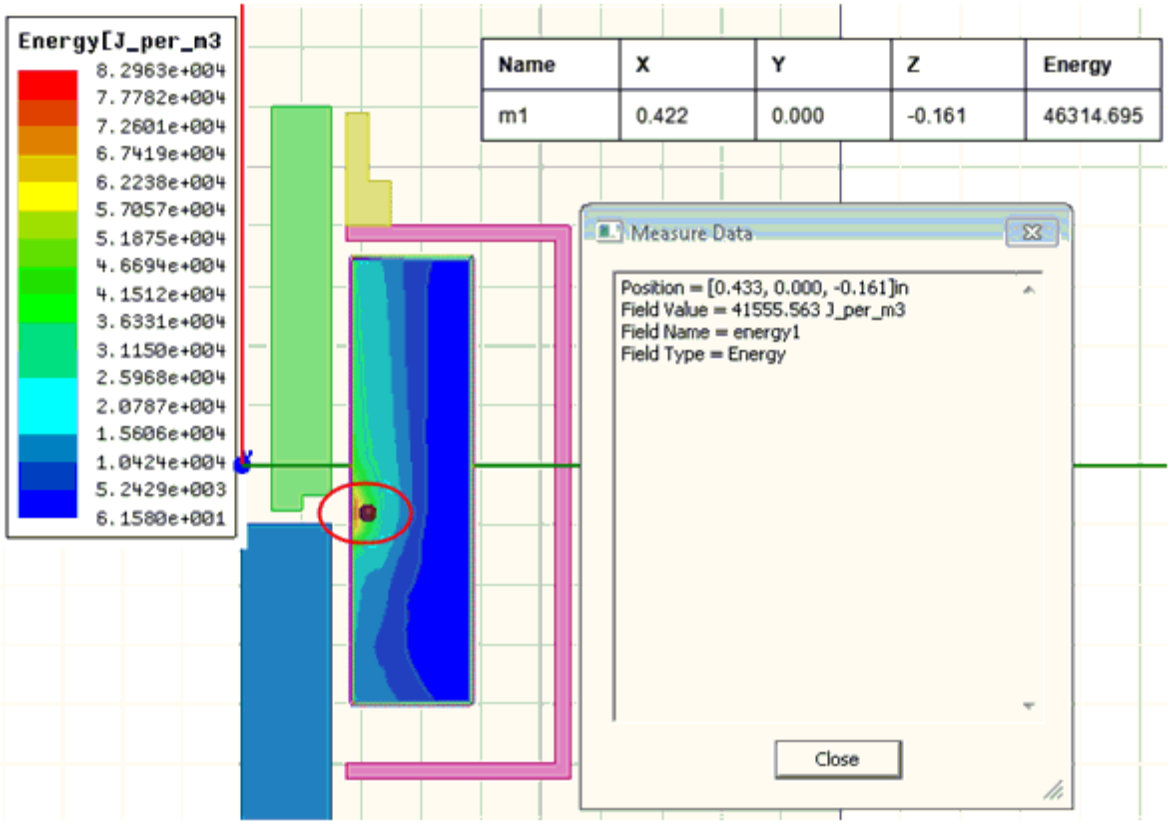
To add one or more field plot markers to a scalar field:

1. On the main menu click **Maxwell3D or Maxwell2D and then select Fields>Fields>Marker>AddMarker**. Alternatively you can right-click anywhere in the

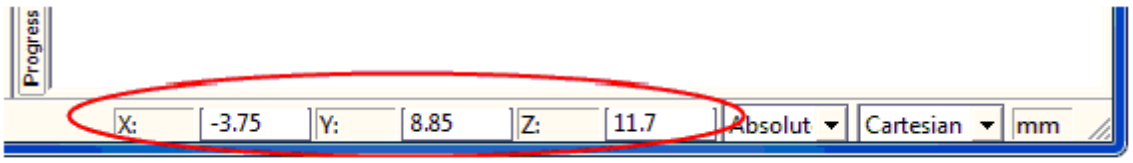
modeler window, or on the Field Overlays folder icon in the Project Manager, and select **Fields>Marker>AddMarker**.

The **Measure Data** dialog box opens and a round dot appears at the tip of the cursor.

2. Drag the dot over the spot on the field overlay plot where you want to add a marker. The Measure Data dialog shows detailed information for the spot currently under the dot.



3. Click the desired point in the field overlay plot to add the marker at that location. Alternatively you can select the position of the marker by entering the values manually in the edit fields at the bottom of the window as shown below.



A table showing the marker coordinates and associated field value is also created and added to the modeler window.

4. Repeat as desired to add additional markers.
5. Press the **Esc** key when finished adding markers.

Deleting a Field Plot Marker

To delete a field plot marker:

1. Click on the marker you want to delete to select it. The row corresponding to the selected marker will be highlighted in the marker table.

   Press and hold the **Ctrl** key and click to select multiple markers.
2. On the main menu click **Maxwell3D or Maxwell2D and then select Fields>Fields>Marker>Delete Marker** to delete the selected marker(s).

   – Alternatively you can right-click anywhere in the modeler window, or on the Field Overlays folder icon in the Project Manager, and select **Fields>Marker>Delete Marker**.

   – You can also simply press the **Delete** key to delete the selected marker(s).

Exporting a Field Plot Marker Table

You can export a field plot marker table to either a comma- or tab-delimited file as follows:

1. On the main menu click **Maxwell3D or Maxwell2D and then select Fields>Fields>Marker>Export Marker Table**.

   Alternatively you can right-click anywhere in the modeler window, or on the Field Overlays folder icon in the Project Manager, and select **Fields>Marker>Export Marker Table**.
2. In the **Export As** dialog box, choose the export format, either **.csv** or **.tab**, and save the file in the desired location.

   The exported file can then be imported into another application such as a spreadsheet.

	A	B	C	D	E
1	Name	X	Y	Z	H[A_per_m]
2	m1	2	22.935	1.517	1234
3	m2	1.517	17.373	1.517	123
4	m3	2.488	23.516	1.517	234
5	m4	2.751	23.828	1.517	1324
6					

Clearing All Field Plot Markers

To clear all field plot markers in the active modeler window do one of the following:

- On the main menu click **Maxwell3D or Maxwell2D and then select Fields>Fields>Marker>Delete All**.
- Alternatively you can right-click anywhere in the modeler window, or on the Field Overlays folder icon in the Project Manager, and select **Fields>Marker>Delete All**.

Deleting all the markers also removes the marker table.

Editing Field Plot Markers

To edit field plot markers:

1. On the main menu click **Maxwell3D or Maxwell2D and then select Fields>Fields>Marker>Edit Marker**.

   Alternatively you can right-click anywhere in the modeler window, or on the Field Overlays folder icon in the Project Manager, and select **Fields>Marker>Edit Marker**.
2. Click on the marker you wish to edit. The row corresponding to the selected marker is highlighted in the marker table. The properties of the marker are displayed in the Properties window.
3. You can change **Position** of the selected marker by editing its coordinate values. The marker table changes to reflect the new value. You can also click the **Color** value bar to select a new color for the marker.
4. Optionally, press the **Delete** key to delete the marker and its corresponding entry in the marker table.
5. When finished editing markers, press **Esc** to exit the marker editing function.

Creating Vector Field Plots

A vector plot uses arrows to illustrate the magnitudes of the x-, y-, and z-components of field quantities. Vector plots can be created on surfaces or volumes.

1. Do one of the following:
   a. To create a vector surface plot, select the faces on which you want to plot the fields.
   b. To create a vector volume plot, select the objects within which you want to plot the fields.
2. Click **Maxwell3D** or **Maxwell2D** and then select **Fields>Fields**.
3. On the **Fields** submenu, click the vector field quantity you want to plot.
4. Follow the procedure for plotting field overlays.

   If you select a vector quantity, you can also check Streamline for the plot.

Note	• Before creating the Streamline plot, select both starting and ending edges (in 2D) or both surfaces (in 3D). • In the **Create Field Plot** dialog box, select **In Volume:** region, which is the region in which the streamlines will appear. Streamlines are outside of objects that are excited (sources). • To show more streamlines after the plot is created, on the **Modify attributes...** dialog **Plots** tab, reduce the **Seeds density** value. If no streamlines appear, reduce this by a factor of 10 (or 100) because the default seeding was too large. See Setting Field Plot Attributes for adjusting the Streamline display parameters, and Setting Fields Reporter Options for setting Streamline defaults. . For exporting Streamline plots, refer to Exporting Field Plots.

Related Topics

Modifying Field Plot Attributes

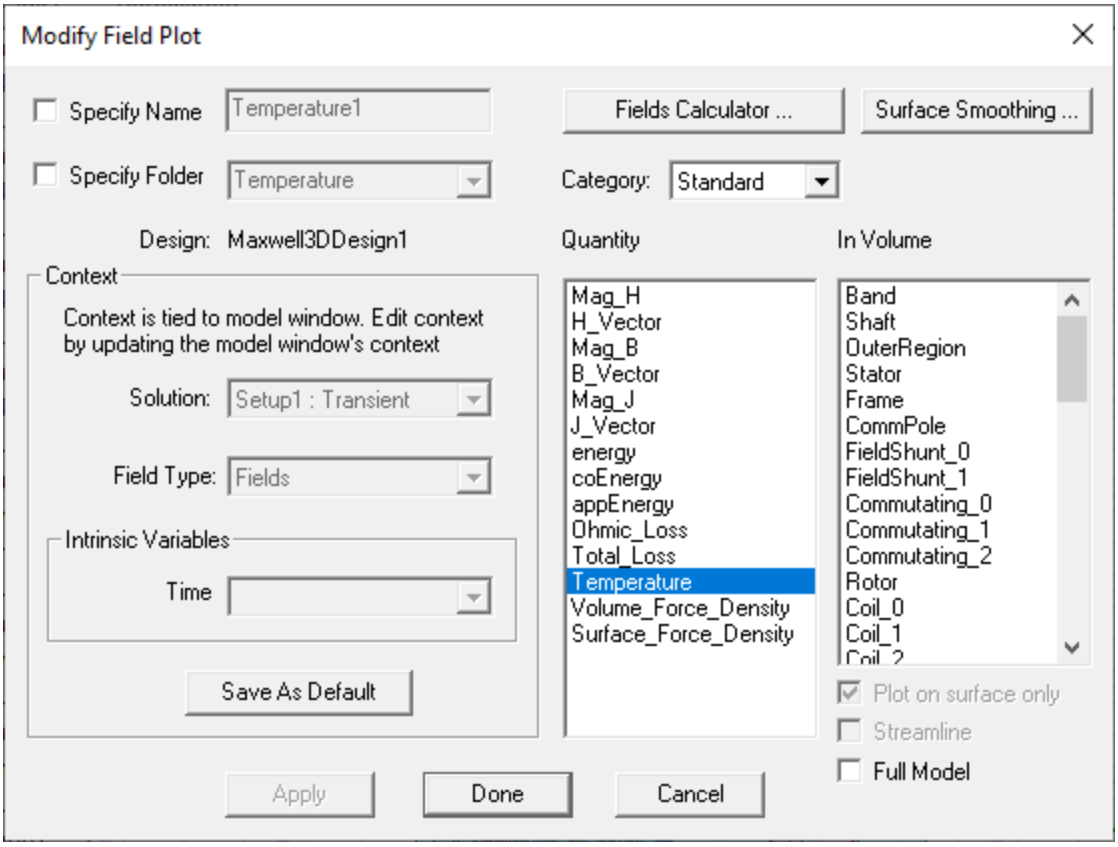
Plotting Field Overlays

Modifying Field Plots

1. Click **Maxwell 3D** or **Maxwell 2D** and then select **Fields>Modify Plots**, or in the Project tree, select the **Field Overlays** icon, right-click, and select **Modify Plots** or use the **M** hotkey.

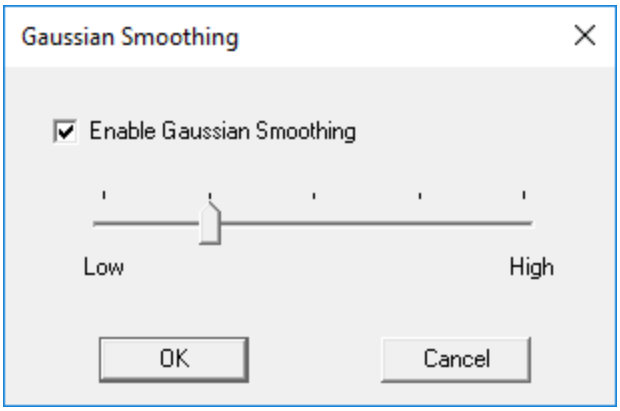
   The **Select Field Plot(s)** dialog box appears.
2. Select the plot you want to modify from the **Select** column, and then click **OK**.

   The **Modify Field Plot** dialog box appears.



Note	Optionally, you can right-click an existing plot listed under the Field Overlays in the Project tree, and select **Modify Plot** to open its **Modify Field Plot** dialog box.

3. Optionally, click a different solution to plot from the **Solution** pull-down list.
4. Optionally, click a different field type (Fields or Noise Vibration) from the **Field Type** pull-down list.
5. Optionally, click the **Specify Name** check box to enable the name field.
6. Optionally, select the **Specify Folder** check box to enable the drop-down menu to select a folder in which to store the plot.
7. Optionally, select a different field quantity to plot from the **Quantity** list.
    - To choose a calculated expression, select **Calculator** from the **Category** pull-down list.
    - To choose a predefined field quantity, select **Standard** from the **Category** pull-down list.
    - For scalar quantities plotted on the surface of a geometry, the **Surface Smoothing** button is enabled. For vector quantities, the **Surface Smoothing** button is disabled. **Surface Smoothing** applies a selected level of smoothing based on the Gaussian Blur algorithm. Clicking **Surface Smoothing** opens the **Gaussian Smoothing** dialog box.



    Select the **Enable Gaussian Smoothing** check box to enable the feature and use the slider to set the degree of smoothing.

    Note: You must click **Apply** in the **Modify Field Plots** dialog box to see the changes.
7. Under **Intrinsic Variables**, select the variable value(s) at which the field quantity is evaluated. Variables available vary depending on the solution type.
8. Select the volume, or region, in which the field is to be plotted from the **In Volume** list. This selection enables you to limit plots to the intersection of a volume and the selected object.
9. For model objects in partial model designs, click the **Full Model** check box to calculate the fields on the full model based on the partial model solution.
10. Click **Apply**, and then click **Done**.

    The field quantity is plotted on the surfaces or within the objects you selected. The new plot is listed in the specified plot folder in the project tree.

    The plot uses the attributes specified in the **Plot Attributes** dialog box.

Related Topics

Setting a Plot's Visibility
Add Trace Characteristics
Setting Field Plot Attributes
Plotting Standard Fields Overlays
Plotting Noise Vibration Overlays

Setting Field Plot Attributes

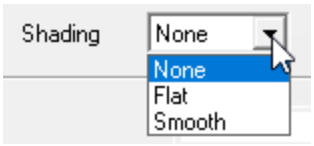
After creating a vector or scalar field overlay on a surface or volume, you can modify its appearance by changing the settings in the **Plot Attributes** dialog box. When you modify the settings for a plot folder, all plots in that folder are modified with the same attributes.

1. Click **Maxwell3D or Maxwell2D and select Fields>Modify Plot Attributes** , or in the Project Manager window, select the **Field Overlays** icon, and select **Modify Attributes**.

   The **Select Plot Folder** window appears.
2. Select the field plot folder you want to modify, and then click **OK**. You can also select the specific plot in the Project tree, and select **Modify Attributes** from the right click menu. (For mesh plots, refer to *Setting Mesh Plot Attributes*.)

   A dialog box with attribute settings for the selected plot appears.
3. Change the desired plot attributes. Under the following tabs in the dialog box, you can control the following attributes:

Color map	The number of colors used and how they are displayed.
Scale	The scale of field quantities.
Marker/Arrow	• The appearance of points (for scalar point plots). • The appearance of arrows (for vector plots).
Deformation Scale	• Select whether to show or hide the deformation and also scale the plot. (Applicable only when stress feedback coupling with ANSYS Structural is enabled.)
Plots (if not streamline)	• The plot selected. • To display or hide the mesh on the plot's surface or volume. • The type of isovalue display (for scalar plots.) • Whether to use Shading (for scalar plots), if lighting is turned on. By default, the Shading is set to None which is equivalent to "Do not use lighting."

	 • The transparency based on solution value. • Whether to add a grid (that is, a mesh overlay), and to set the grid color. • Specify the plot resolution as **Coarse**, **Normal**, **Fine**, or **Very Fine**. This affects the use of memory for animating plots. For large plots with more frames to animate, use **Coarse** or **Normal** to reduce memory requirements and improve performance. For smaller plots with few frames, if higher resolution is required, use **Fine** or **Very Fine**. • The spacing of arrows (for vector plots).
Plots (if streamline is checked)	• The plot selected • The linestyle as solid or cylinder from drop-down menu. • Line width, specified using a slider. • Whether to show marker on streamline. • Seeds density spacing. This affects the number of stream lines used to represent the quantity in the plot. Moving the slider to the left decreases the spacing and increases the number of stream lines. Moving the slider to the right increases the spacing and decreases the number of lines used to represent the quantity. • Min. and Max. values represented.

4. Optionally, click **Save as default** if you want the tab's settings to apply to field overlay plots created after this point.
5. Select **Real time mode** if you want the changes to take effect immediately in the view window. If this option is cleared, click **Apply** when you want to see the changes.
6. Click **Close** to dismiss the dialog box.

Related Topics

Setting a Plot's Visibility
Plotting the Mesh
Setting Mesh Plot Attributes
Plotting Field Overlays
Modifying the Lighting

Modifying Field Plot Colors

1. Click **Maxwell3D or Maxwell2D and select Fields>Modify Plot Attributes** , or in the Project tree, right click on the Field Overlays icon and select **Modify Plots** from the short-cut menu,.

   The **Select Plot Folder** window appears.
2. Select the plot you want to modify, and then click **OK**.

Note	All plots in the selected folder are modified.

   A dialog box with attribute settings for the selected folder appears.
3. Click the **Color map** tab.
4. Select one of the following color types:

Uniform	Field quantities are plotted in a single color. Select the plot color from the **Color** palette.
Ramp	Field quantities are plotted in shades of a single color. Select the plot color from the **Color** palette. The shade of the color corresponds to its field value.
Spectrum	Field quantities are plotted in multiple colors. Select a color spectrum from the pull-down list. Each field value is assigned a color from the selected spectrum.

   You can choose **Save as Default**, if you want to use the current settings.

   Select **Real time mode** if you want these, or subsequent changes to take effect immediately in the view window.

   If this option is cleared, click **Apply** when you want to see the changes.
5. Click the **Scale Tab**.
6. In the **Num. Divisions** field, enter the number of colors to use in the plot.

   You can choose **Save as Default**, if you want to use the current settings.

   Select **Real time mode** if you want the changes to take effect immediately in the view window.

   If this option is cleared, click **Apply** when you want to see the changes.
7. Click **Close** to dismiss the dialog box.

Related Topics

Setting a Plot's Visibility

Setting the Color Key Visibility

Moving the Color Key

Setting the Color Key Visibility

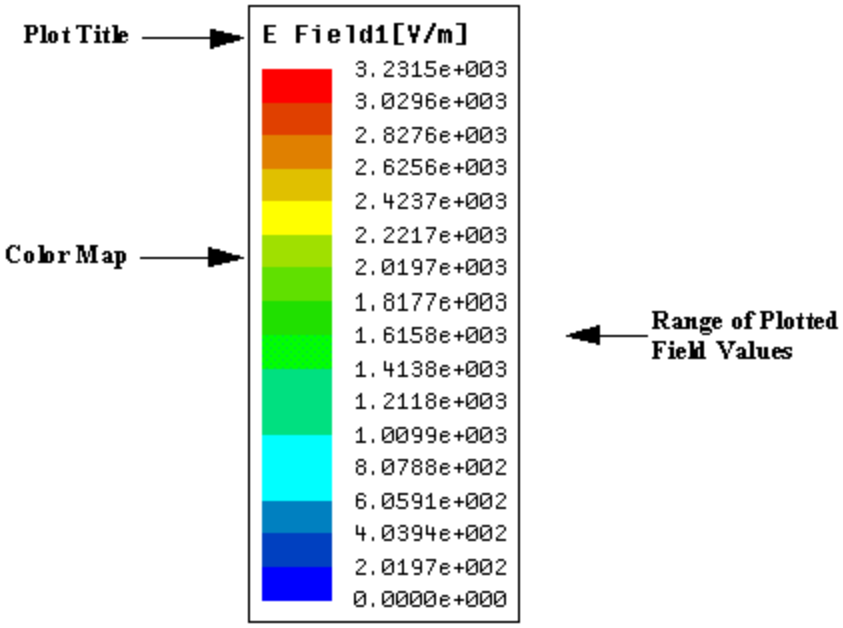
The color key (shown below) displays the range of plotted field values for a field overlay plot. It displays the colors that correspond to the range of field values on the plot.

1. Click **View**>**Active View****Visibility**  .

   The **Active View Visibility** dialog box appears.
2. Click the **Color Keys** tab.
3. In the **Visibility** column, select the field overlay or mesh plots in which you want to display the color key. Clear the plots for which you want to hide the color key from view.
4. Click **Done** to close the dialog box.

   Alternatively, to hide the color key, right-click on the color key in the view window, and then click **Hide** from the shortcut menu.

   Only the color keys in the selected plots are visible.



Related Topics

Modifying Field Plot Colors

Moving the Color Key

Moving the Color Key

Click on the active field overlay plot's color key and drag it to a new location.

Related Topics

Setting the Color Key Visibility

Modifying the Field Plot Scale

To change how field quantities are scaled on the field overlay plot:

1. Click **Maxwell3D or Maxwell2D and select Fields>Modify Plot Attributes** or in the Project tree, right click on the Field Overlays icon and select **Modify Plots** from the short-cut menu.

   The **Select Plot Folder** window appears.
2. Select the plot folder you want to modify, and then click **OK**.

Note	All plots in the selected folder are modified.

   A dialog box with attribute settings for the selected folder appears.
3. Click the **Scale** tab.
4. Optionally, to change the number of divisions in the field plot scale, set the **Num. Division** field to a new value. You can click **Save as Default**, if desired.
5. Select one of the following scale options:

Auto	The full range of field values will be plotted on the selected surface or volume. Selecting **Auto** enables the **Auto Scale Options** and disables the **Min** and **Max** fields. By default, precision is not limited and auto-min is the actual computed min on the plotted geometry.
Use Limits	Only the field values between the minimum and maximum values will be plotted. Field values below or above these values will be plotted in the colors assigned to the minimum or maximum limits, respectively. Selecting **Use Limits** enables the **Min** and **Max** fields and disables the **Auto Scale Options**. Field values have a precision of at most 6 decimal places (field solution files are saved in floating precision), so Min/Max numbers are displayed to this precision.
Specify Values	This enables a **Scale Values** button.

a. If you selected **Use Limits**, enter the lowest field value to be plotted in the **Min.** box and the highest field value to be plotted in the **Max.** box.
b. If you selected **Auto**, the **Auto Scale Options** are enabled. You should only change this for cases where auto-min is a small number. Use the **Limits Max/Min precision to** check box to enable setting the drop down menu for the precision limit. The auto-min is the greater of the following:
   - Actual computed Min
   - Max/pow(10, num digits of field precision)
c. If you selected **Specify Values**, you can click the **Scale Values** button. This opens a dialog with an editable, scrollable list of the current scale values. To apply the changes you make, click the **OK** button. To close the dialog without make changes, click **Cancel**.
d. Optionally, when dB is checked, the dB scale is used for the plot.

6. Optionally, use the **Units** drop down menu to select the default unit of measure for the plot.

   The units specified here appear on the Color map for the fields plot, and for the properties dialog for the field quantities.

7. If you selected **Auto** or **Use Limits** and without have **dB** checked, you can select one of the following options:

Linear	Field values are plotted on a linear scale.
Log	Field values are plotted on a logarithmic scale. If field plots have negative and positive values and when auto-scale is selected, the log-scale choice automatically sets the Min value as the Max/Min Ratio. (If field plots have all negative values, **Log** is not allowed.)

8. Specify the **Number Format** for the plot field values as Auto, Scientific or Decimal. You can also specify Width and Precision if Number Format is Scientific or Decimal.
9. Select **Real time mode** if you want the changes to take effect immediately in the view window.

   If this option is cleared, click **Apply** when you want to see the changes.

10. Optionally, you can use the **Save As Default** button to save the following to registry:
    - Whether to limit field precision,
    - The number of digits of field precision,
    - Whether to use log/linear scale.
    - Whether to use dB.

    Auto scale is always be default for new plots. For scalar-in-volume plots, iso-surface (rather than cloud) is the default display

11. Click **Apply**, and then click **Close** to dismiss the window.

Modifying Vector Field Plot Arrows

To change the appearance of a vector field plot's arrows:

1. Click **Maxwell3D or Maxwell2D and selectFields>Modify Plot Attributes** .

   The **Select Plot Folder** window appears.
2. Select the plot folder you want to modify, and then click **OK**.

Note	All plots in the selected folder are modified.

   A dialog box with attribute settings for the selected folder appears.
3. Click the **Marker/Arrow** tab.
4. Under **Arrow options**, select one of the following arrow types from the **Type** pull-down list:

Line	The arrows are displayed as 2D/flat.
Cylinder	The arrow tails are displayed as cylinders. The arrowheads are displayed as 3D/round.
Umbrella	The arrow tails are displayed as 1D lines. The arrowheads are displayed as 3D/round.

5. Use the **Size** slider to increase (move to the right) or decrease (move to the left) the length and dimensions of the arrows. The arrows are resized relative to the size of the model geometry.
6. Select **Map size** to scale the size of the arrows to the magnitude of the field quantity being plotted.
7. Select **Arrow tail** to include tails on all arrows.
8. Click the **Plots** tab.

Note	Maxwell plots arrows on a grid that is superimposed on the surface or object you selected for the plot.

9. Under **Vector plot**, use the **Spacing** slider to increase (move to the right) or decrease (move to the left) the distance between arrows (grid points.)
10. If you want the arrows to be spaced equally, select **Uniform**, and enter values in the Min and Max boxes.
11. Select **Real time mode** if you want the changes to take effect immediately in the view window. If this option is cleared, click **Apply** when you want to see the changes.
12. Click **Apply**, and then click **Close** to dismiss the window.

Setting the Mesh Visibility on Field Plots

To display or hide the mesh on field plots or to change the mesh's color:

1. Click **Maxwell3D or Maxwell2D and selectFields>Modify Plot Attributes** .

   The **Select Plot Folder** window appears.
2. Select the plot folder you want to modify, and then click **OK**.

Note	All plots in the selected folder are modified.

   A dialog box with attribute settings for the selected folder appears.
3. Click the **Plots** tab.
4. Select **Add grid** to display the mesh.
5. Optionally, select a color for the mesh from the **Color** palette.
6. Select **Real time mode** if you want the changes to take effect immediately in the view window.

   If this option is cleared, click **Apply** when you want to see the changes.
7. Click **Apply**, and then click **Close**.

Related Topics

Plotting the Mesh

Setting a Plot's Visibility

Modifying Scalar Field Plot Isovalues

1. Click **Maxwell3D** or **Maxwell2D** and select **Fields>Modify Plot Attributes** .

   The **Select Plot Folder** window appears.
2. Select the plot folder you want to modify, and then click **OK**.

Note	All plots in the selected folder are modified.

   A dialog box with attribute settings for the selected folder appears.
3. Click the **Plots** tab.
4. If the plot is a scalar surface plot, do the following:
   a. Select one of the following isosurface display types from the **IsoValType** pull-down list:

Line	Lines are drawn along the isovalues.
Fringe	Color is constant between isovalues.

Tone	Color varies continuously between isovalues.
Gourard	Color varies continuously across the plot.

b. Optionally, if you selected **Fringe** or **Tone**, select **Outline** to add a border line between isovalues.

c. If you select **Map transp.**, the transparency of field values increases as the solution values decrease. Use the **Map transp.** slider to increase (move to the right) or decrease (move to the left) the transparency of the plot.

d. For shading, if you are using Lighting, you can adjust the **Shading** as:

None	Equivalent to selecting **Do not use lighting** on the Lighting Properties dialog box.
Flat	Provides a flat rendering of the field plot.
Smooth	Provides a smooth rendering of the field plot.

5. If the plot is a scalar volume plot, do the following:

a. Select one of the following display types:

IsoValSurface	Color is drawn on the isovalues.
Cloud	Field values are represented by points that illustrate the spatial distribution of the solution. The higher the solution value, the greater the cloud density.

b. Optionally, if you selected **Cloud**, use the **Particle spacing** slider to increase or decrease the number of points that represent the density on the volume. The Min and Max fields allow you to set the limits for the Particle spacing slider.

c. Optionally, if you selected **Cloud**, enter a size for particles in the clouds in the **Particle size** box.

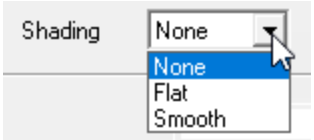
d. For shading, if you are using Lighting, you can adjust the **Shading** as:

None	Equivalent to **Do not use lighting** on the Lighting Properties dialog box.
Flat	Provides a flat rendering of the field plot.
Smooth	Provides a smooth rendering of the field plot.

8. Select **Real time mode** if you want the changes to take effect immediately in the view window. If this option is cleared, click **Apply** when you want to see the changes.
9. Click **Close** to dismiss the window.

Modifying Scalar Field Plot Shading

1. Click **Maxwell** > **Fields**> **Modify Plot Attributes**.
   The **Select Plot Folder** window appears.
2. Select the plot folder you want to modify, and then click **OK**.
   All plots in the selected folder will be modified.
   A dialog box with attribute settings for the selected folder appears.
3. Click the **Plots** tab.
4. If the plot is either a scalar surface plot or a scalar volume plot, do the following:
   a. Select one of the following from the Shading drop-down menu:



None	Equivalent to selecting **Do not use lighting** on the Lighting Properties dialog box.
Flat	Provides a flat rendering of the field plot.
Smooth	Provides a smooth rendering of the field plot.

5. Select **Real time mode** if you want the changes to take effect immediately in the view window.
   If this option is cleared, click **Apply** when you want to see the changes.
6. Click **Close** to dismiss the window.

Mapping Scalar Field Plot Transparency to Field Values

1. Click **Maxwell3D or Maxwell2D and selectFields**>**Modify Plot Attributes** .
   The **Select Plot Folder** window appears.
2. Select the plot folder you want to modify, and then click **OK**.

Note	All plots in the selected folder are modified.

   A dialog box with attribute settings for the selected folder appears.
3. Click the **Plots** tab.
4. Use the **Map transp.** slider to increase (move to the right) or decrease (move to the left) the transparency of the plot.

- If you select **Map transp.**, the transparency of field values increases as the solution values decrease.

5. Select **Real time mode** if you want the changes to take effect immediately in the view window.

   If this option is cleared, click **Apply** when you want to see the changes.
6. Click **Apply**, and then click **Close** to dismiss the window.

Modifying Markers on Point Plots

For scalar point plots, a marker is used to represent a field quantity at a selected point. (For vector point plots, arrows are used.) Modify the shape and size of markers in the plot attributes window.

1. Click **Maxwell3D or Maxwell2D and selectFields>Modify Plot Attributes**.

   The **Select Plot Folder** window appears.
2. Select the plot folder you want to modify, and then click **OK**.

Note	All plots in the selected folder are modified.

   A dialog box with attribute settings for the selected folder appears.
3. Click the **Marker/Arrow** tab in the plot attributes window.
4. Under **Marker options**, select one of the following marker types to represent the field quantity at the point:
   - **Sphere**
   - **Box**
   - **Tetrahedron**
   - **Octahedron**
5. Use the **Size** slider to increase (move to the right) or decrease (move to the left) the size of the marker.
6. Select **Map size** to scale the size of the marker to the magnitude of the quantity being plotted.
7. Select **Real time mode** if you want the changes to take effect immediately in the view window. If this option is cleared, click **Apply** when you want to see the changes.
8. Click **Apply**, and then click **Close**.

Related Topics

Drawing a Point

Viewing Data Markers

The difference between two markers can be viewed on a plot.

1. Click **Report 2D>Data Marker**.
2. Click the first point on the graph.
3. Hold-and-drag the mouse and release at the second point.

Under the XY plot, you can see the difference between the two marker points.

Modifying Line Plots

Field quantities can be plotted directly on a line object. Scalar quantities are plotted as 3D color-shaded lines. Vector quantities are plotted as arrows that are based on the line.

To modify the appearance of line plots:

1. Click **Maxwell3D or Maxwell2D and selectFields>Modify Plot Attributes** .

   The **Select Plot Folder** window appears.
2. Select the plot folder you want to modify, and then click **OK**.

Note	All plots in the selected folder are modified.

   A dialog box with attribute settings for the selected folder appears.
3. Click the **Plots** tab.
4. Select one of the following isosurface display types from the **IsoValType** pull-down list:

Fringe	Color is constant between isovalues.
Tone	Color varies continuously between isovalues.
Gourard	Color varies continuously across the plot.

5. Select one of the following styles for the line object from the **Line style** pull-down list:

Cylinder	The line object is shaped like a cylinder.
Solid	The line object is a 3D solid.
Dash-Dash	The line object is represented by dashed black line segments.
Dot-Dot	The line object is represented by a series of dots.
Dash-Dot	The line object is represented by a series of alternating dashed black line segments and dots.

6. Use the **Line width** slider to increase (move to the right) or decrease (move to the left) the thickness of the line.

7. By default, a polyline object is divided into 100 equally spaced points for post processing. To modify the number of points on the line, type a new value in the **Number of points** box.
8. Select **Real time mode** if you want the changes to take effect immediately in the view window. If this option is cleared, click **Apply** when you want to see the changes.
9. Click **Apply**, and then click **Close**.

Related Topics

Drawing a Polyline

Setting a Plot's Visibility

To display or hide a field overlay or mesh plot from view in the **3D Modeler** window:

1. Click **View>Visibility>Active View Visibility**.

   Alternatively, you can select the **Active View Visibility** icon  on the **View** ribbon.

   The **Active View Visibility** dialog box appears.
2. Click the **FieldsReporter** tab.
3. In the **Visibility** column, select the field overlay or mesh plots you want to display. Clear the plots you want to hide from view.
4. Click **Done**.

   Only the selected plots are visible.

Saving a Field Overlay Plot

Field overlay and mesh plots are saved in the project file (.adsn); however, you can save a plot to a Maxwell Field Plot File format (.dsp) and then open it in Maxwell.

To save field overlay or mesh plot data to a .dsp file:

1. In the project tree, click the plot you want to export.

2. Click **Maxwell3D or Maxwell2D and selectFields>Save as** .

   The **Select Field Plot(s)** dialog box appears.
3. Select the plots you want to export by checking the **Select** box, and then click **OK**.

   A file browser appears.
4. Make sure that **Field Plot Files (.dsp)** is the selected file type.
5. Specify the name of the .dsp file and the location in which to save it.
6. Click **Save**.

The plot is exported to the specified .dsp file.

The file you created can be opened in Maxwell version 9 and later. Simply click **Maxwell3D or Maxwell2D and selectFields>Open**.

Opening a Field Overlay Plot

To open a field overlay or mesh plot that you have saved to Maxwell Field Plot File format (.dsp) in Maxwell version 9 and later:

1. Click **Maxwell3D or Maxwell2D and selectFields>Open** .

   A file browser **Open** dialog box appears.
2. Make sure that **Field Plot Files (.dsp)** is the selected file type in the **Files of type** pull-down list.
3. Browse to the location of the .dsp file you want to open, and then click the file name.
4. Click **Open**.

   The plot appears in the view window and is listed under **Field Overlays** in the project tree.

Deleting a Field Overlay Plot

1. Click **Maxwell3D or Maxwell2D and selectFields>Delete Plot** .

   The **Delete Plots** dialog box appears.
2. Select the plot(s) you want to delete by checking the **Delete** check box in the appropriate row.
3. Click **OK**.

   The selected plots are deleted.

Alternatively, click the plot in the project tree that you want to delete, and then press **Delete**.

Setting Field Plot Defaults

Each new field plot uses the default plot settings specified in the **Set Plot Defaults** dialog box.

To modify the default plot settings:

1. If a plot folder has not been created, click **Field Overlays** in the project tree.

2. Click **Maxwell3D or Maxwell2D and selectFields>Set Plot Defaults** .

The **Set Plot Defaults** dialog box appears.

3. Select the solution to plot from the **Solution** pull-down list.
4. Select the plot folder in which new plots will be stored from the **Plot Folder** pull-down list. Select from one of the following options:

New Folder	Each new plot is stored in a separate folder in the project tree.
Automatic	Each new plot is stored in a folder determined by Maxwell as the most appropriate based on the plotted field quantity. For example, all surface magnitude E plots are stored in the same folder.
An existing folder	Select the existing folder in which you want to store new plots.

Note	Plots stored in the same folder use the same color key. The **Auto** scale setting is based on the maximum field solution value present in a plot.

5. Under **Intrinsic Variables**, specify the frequency and phase angle at which the field quantity is evaluated.
6. Click **OK**.

Renaming a Plot

To rename a plot, do the following in the project tree:

1. Under **Field Overlays**, expand the field type, and right-click the specific plot you want to rename.

   A shortcut menu appears.
2. Select **Rename**.

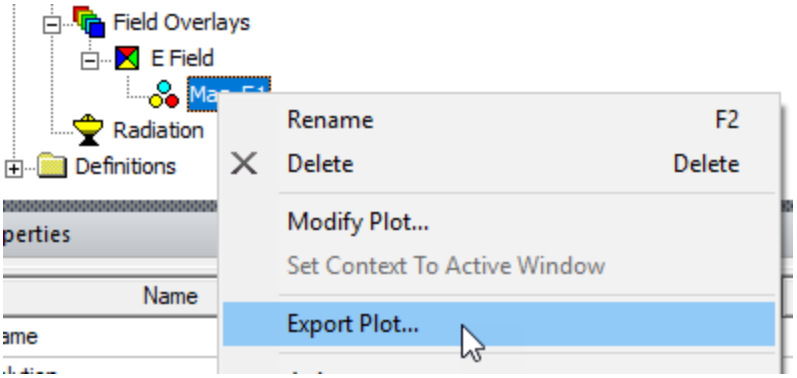
   The text becomes editable in the project tree.
3. Type the new name, and press **Enter**.

Exporting Field Plots

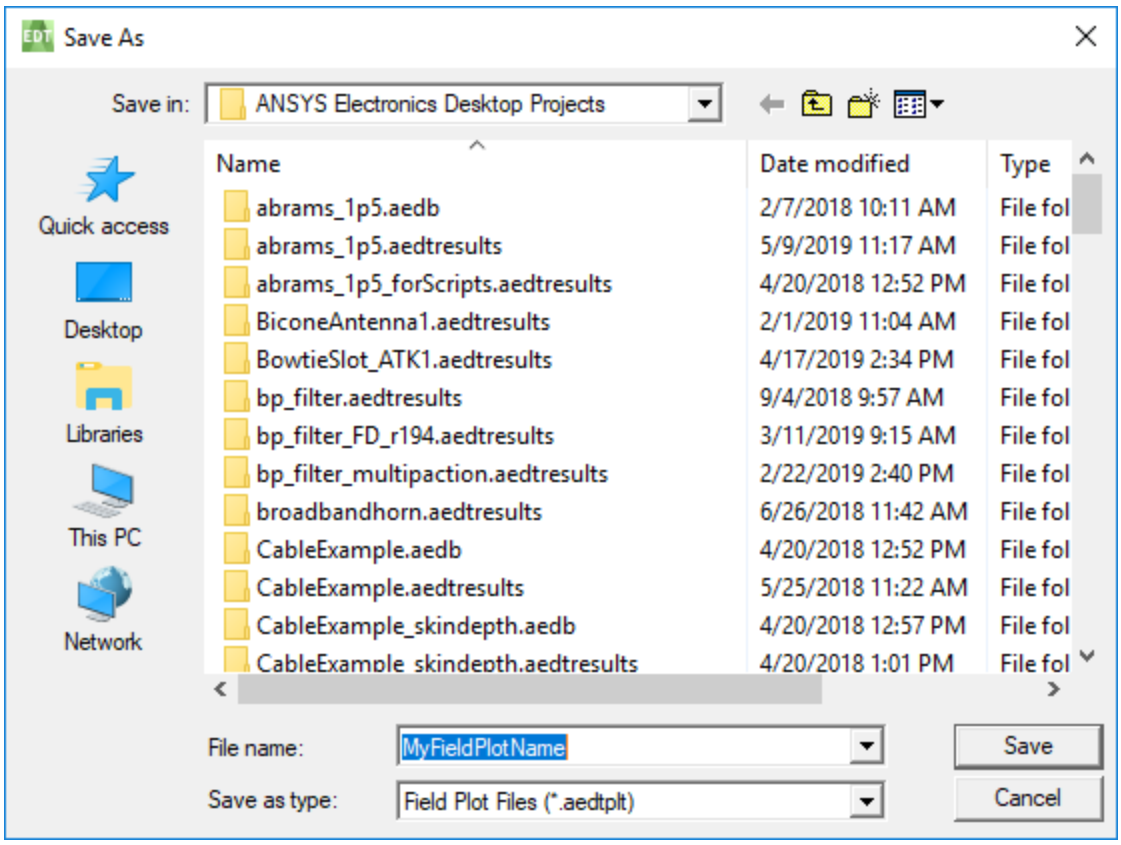
The **Export Plot...** command lets you export a field plot to a format that supports full data precision without the need to go through the Fields Calculator.

1. Create the field plot you want to export.

2. In the **History tree**, right click on the field plot icon and in the shortcut menu, click **Export Plot...**.



The **Save As** dialog box appears.



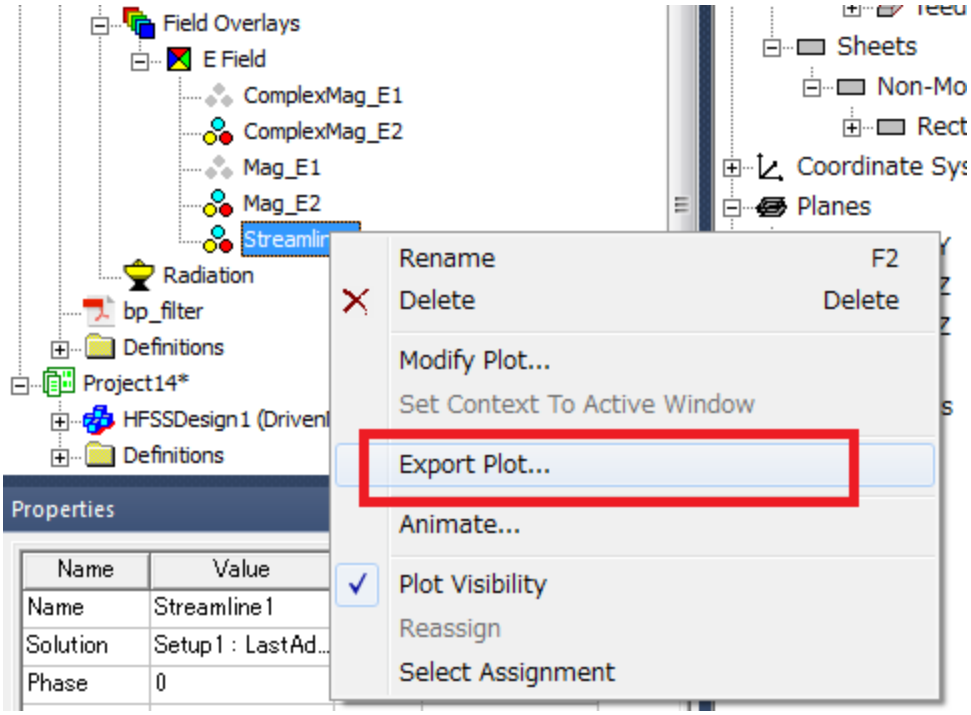
3. Specify the directory to **Save in**, the **File name** and use the **Save as type** as Field Plot Files (*.aedtplot) or Ensight Case Gold Files (*.case).

   If the Plot is a Streamline plot you can choose Streamline files (*.fldplt).

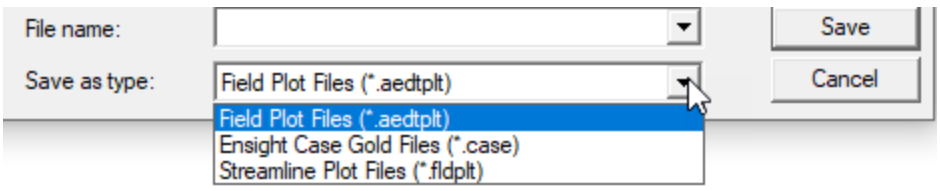
Note	• Before creating the Streamline plot, select both starting and ending edges

	(in 2D) or both surfaces (in 3D). • In the **Create Field Plot** dialog box, select **In Volume:** region, which is the region in which the streamlines will appear. Streamlines are outside of objects that are excited (sources). • To show more streamlines after the plot is created, on the **Modify attributes...** dialog **Plots** tab, reduce the **Seeds density** value. If no streamlines appear, reduce this by a factor of 10 (or 100) because the default seeding was too large. Refer to Setting Field Plot Attributes for adjusting the Streamline display parameters, and Setting Fields Reporter Options for setting Streamline defaults.

You can export a Streamline plot in .fldplt format by right-clicking on the plot in the Project tree, and selecting **Export Plot...**



This opens a dialog for you to specify a plot name and location.



The Field Plot is exported to the file format you specified.

Field Plot Data Export Format (*.aedtplt)

A partial sample file with 2 drawings:

```
# Ansys Electronics Desktop 2022.1
# Field plot export file (*.aedtplt), version 1.0
Number of drawing: 2
$begin Drawing_1
HasCurvElem=false
BoundingBox(-4.4999998807907104e-01, 4.4999998807907104e-01,
2.0000000000000000e+00, 2.0000000000000000e+00,
0.0000000000000000e+00, 4.0000000596046448e-01)
Elements(221, 98, 2, 3, 3, 0, 6, 18, 113, 11, 137, 112, 15,...)
Nodes(4.5000000000000001e-01, 2.0000000000000000e+00,
0.0000000000000000e+00, -4.5000000000000001e-01,
2.0000000000000000e+00, 0.0000000000000000e+00,
4.5000000000000001e-01, 2.0000000000000000e+00,
4.0000000000000002e-01, -4.5000000000000001e-01,
2.0000000000000000e+00, 4.0000000000000002e-01, -
7.3360628685363773e-03, 2.0000000000000000e+00,
2.0484394892748192e-01, 0.0000000000000000e+00,
2.0000000000000000e+00, 0.0000000000000000e+00, ...)
ElemSolution(1.7500478913132198e+00, 3.5599710688669930e+00, 6,
1.9736298171649147e+00, 2.2454764621743504e+00,
2.6385813406790546e+00, 2.2343980354519912e+00,
2.5920843683349752e+00, 2.5976178259591762e+00, ...)
ElemSolutionMinMaxLocation(-9.3167998430411994e-05,
5.0799999999999998e-02, 6.4115181513790202e-03,
1.1429999999999999e-02, 5.0799999999999998e-02,
1.0160000000000001e-02)
$end Drawing_1
$begin Drawing_2
HasCurvElem=false
BoundingBox(-4.4999998807907104e-01, 4.4999998807907104e-01, -
2.0000000000000000e+00, 2.0000000000000000e+00,
4.0000000596046448e-01, 4.0000000596046448e-01)
Elements(204, 102, 1, 3, 0, 0, 3, 1, 103, 2, ...)
Nodes(4.5000000000000001e-01, 2.0000000000000000e+00,
4.0000000000000002e-01, 4.4090909090909097e-01,
1.9595959595959593e+00, 3.9999999999999997e-01,
4.3181818181818193e-01, 1.9191919191919191e+00,
3.9999999999999997e-01, ...)
```

```
ElemSolution(4.8316948198374815e-01, 7.8381454718992716e+00, 3,
3.5599710688669930e+00, 3.5599710688669930e+00,
3.5599710688669930e+00, ...)
ElemSolutionMinMaxLocation(1.7318181818181887e-03,
7.6969696969697125e-03, 1.0160000000000001e-02,
0.0000000000000000e+00, -1.4745149545802860e-17,
1.0160000000000001e-02)
$end Drawing_2
```

File format description

The first line starts with the “#” symbol and continues with the product name and its version that is used to produce this file. The 2nd line also starts with the “#” symbol and continues with the file format version.

The next line shows the number of drawings in this file. The above example shows 2 drawings. Each drawing has two big data blocks (the **mesh data block** and **field data block**). Each drawing starts with **$begin Drawing_***** and ends with **$end Drawing_***** (where ***** should be replaced with the drawing index, starting from 1).

Mesh data block

This contains all the mesh information about the current drawing include element type, number of nodes of each element, node coordinates, and element node connectivity. At the beginning of the block, it shows if it has curve element and then lists the bounding box of this mesh. The bounding box format is **BoundingBox(lowerX, lowerY, lowerZ, upperX, upperY, upperZ)**. These are double values.

Next comes the elements part, starting with **Elements(**. The first integer is the total number of nodes for this mesh, and the next is the total number of elements. After that, it repeats the following information for each element: element type, reserved integer 1, reserved integer 2, reserved integer 3, number of nodes for this element, node index 1, node index 2 till the number of nodes for this element. All these numbers are integers. The **Elements** part ends with a **)**. The list of element types is listed in the Element types section below.

Element type description	Element type value
point	0
line	1
triangle	2
quad	3
tetradedron	4
pyramid	5
wedge	6
hexahedron	7

The last part of the mesh data block is the nodes part. It contains the coordinates (x,y,z) of all the nodes for this mesh. It starts with **Nodes(** and continue with a list of tuples of 3 double values which are the x, y, z coordinates of the first, 2nd, last nodes and it ends with **)**.

Field data block

This contains two parts. The first part includes node field data of each element. Node field data could either be scalar or vector data which means either it has 1 component or it has 3 component (x, y, z).

This part starts with **ElemSolution(**, the next two double values are the minimal and maximal field value of this drawing. The next integer shows the total number field data for each element. It should be a multiple of the number of nodes of each element (e.g., if field data is vector data, then it should be 3 times of the number of nodes of each element). For vector data, each element should have data like (**node1_x, node1_y, node1_z, node2_x, node2_y, node2_z, …**). This part ends with **)**.

The second part shows two location points. It starts with **ElemSolutionMinMaxLocation(**. The first 3 double values are the (x,y,z) coordinates of the location where the minimal field value is located, while the next 3 numbers are for the maximal field value. This part ends with **)**.

For 2nd order element node index convention, please refer to http://www.vki.com/2013/Support/docs/vistools-1.html#1.2. Lagrange Parabolic is used.

Related Topics

Creating Animations

Using the Fields Calculator

The Fields calculator enables you to perform computations using basic field quantities. The calculator computes derived quantities from the general electromagnetic field solution; writes field quantities to files, locates maximum and minimum field values, and performs other operations on the field solution.

The calculator allows you to define a series of calculations to be performed on the field solution but does not actually perform the computations until data is required, for a field plot as an example. This makes it more efficient, saving computing resources and time. You can do all the calculations without regard to data storage of all the calculated points of the field. It is generally easier to do all the calculations first, then plot the results.

Note	In **Maxwell2D**, the Fields Calculator accounts for the difference between XY and RZ models when handling data and results. For example, the integral command will perform an XY integral on the model assuming a 1 meter depth, while for an RZ model it will perform an integral on the 360 degree equivalent volume.

Related Topics

Opening the Fields Calculator

Context Area

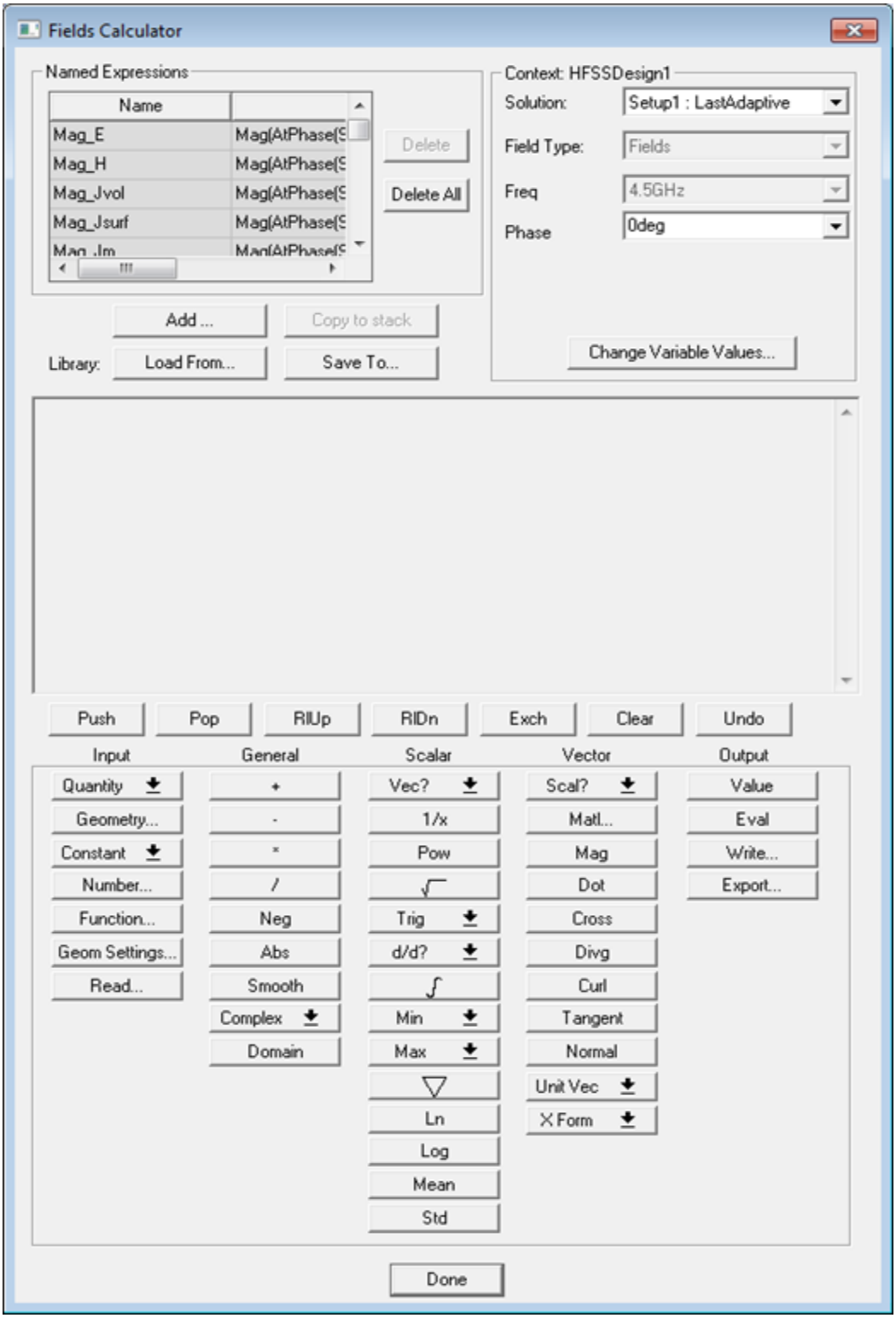
Opening the Fields Calculator

To open the Fields Calculator, do one of the following:

- Click **Maxwell3D or Maxwell2D and select Fields>Calculator**  .
- Right-click **Field Overlays** in the project tree, and then click **Calculator** on the shortcut menu.

The **Fields Calculator** dialog box appears.

To view information on a command or screen area, click over the button or screen area on the illustration below.



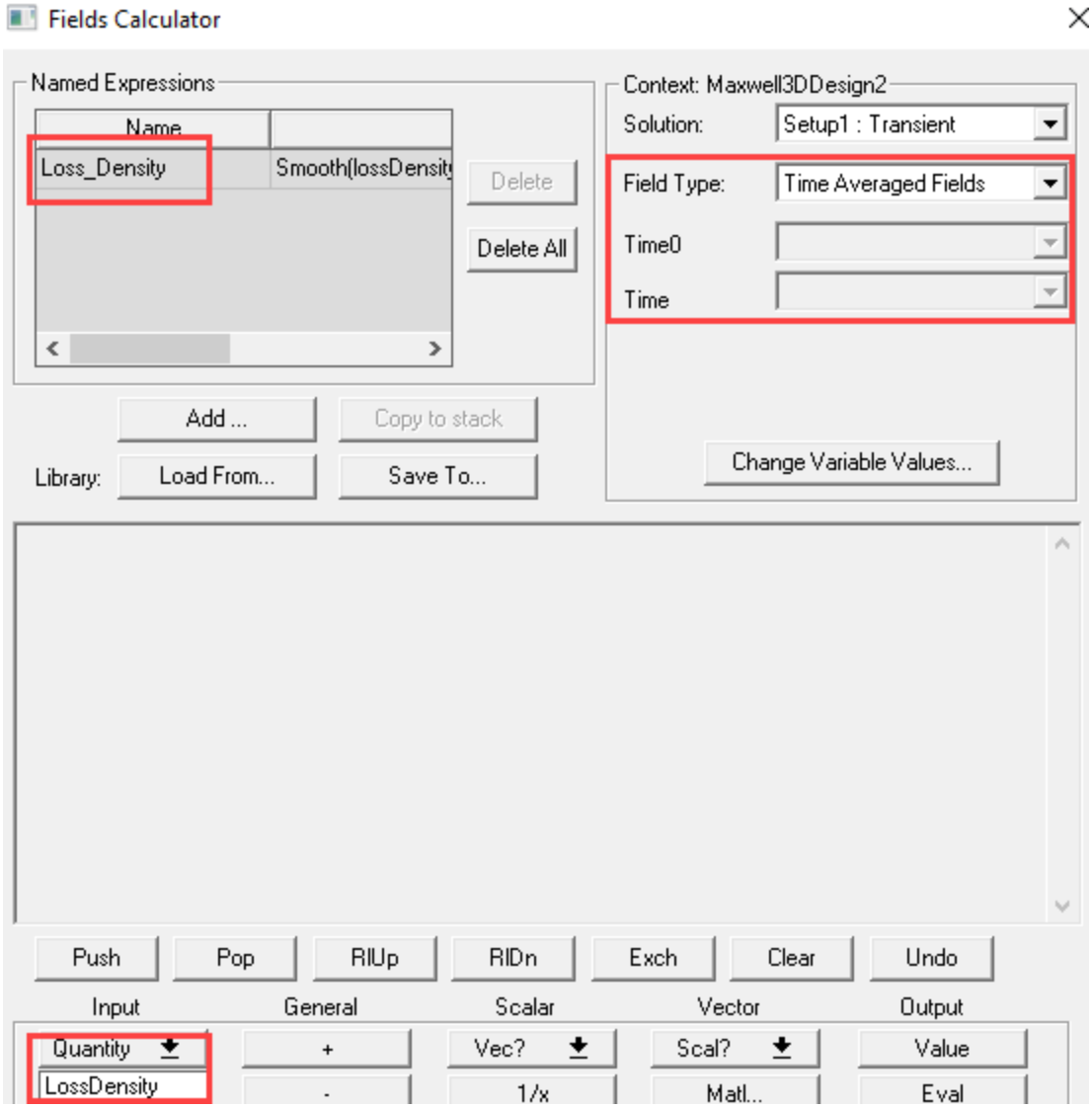
Context Area

The panel at the upper right of the window identifies the context to be used for the calculations. The top line identifies the design. Depending on the design, text entry boxes allow you to select a **Solution**, **Field Type**, **Phase**, or **Time**.

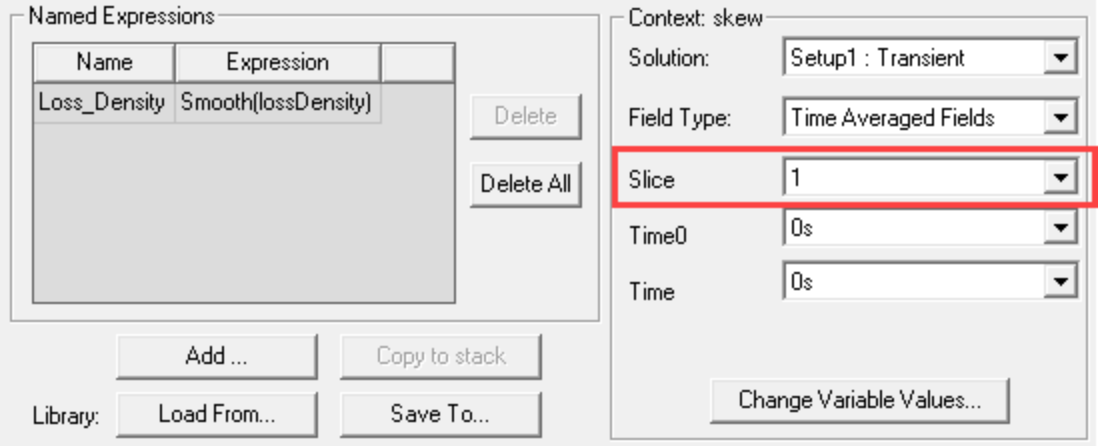
Note	The Field Type here is not related to the edit sources. This is a general term among Ansys Electromagnetics products (HFSS, Maxwell, Q3D, Icepak, and Mechanical). Some products have more than one field type for different solution types. If only one Field Type is available the box is grayed out.

For Maxwell 2D and 3D transient solutions, the field type can be either **Fields** or **Time Averaged Fields**. Selecting **Time Averaged Fields** adds an additional **Time0** field, which is used when calculating time averaged losses. Selecting **Time Averaged Fields** also adds **Loss_Density** as a **Named Expression**, and **LossDensity** as a **Quantity command** selection.

The **Change Variable Values** button opens a **Set Variable Values** dialog box. By default it has **Use Nominal Design** checked. Unchecking the box lets you select another variable value. **OK** the dialog to accept the selection.



For Maxwell 2D transient solutions in which **Use Skew Model is enabled on the Model Settings tab**, an additional **Slice** field displays, which allows you to select the skew model slice to use for fields calculations.



Related Topics

Opening the Fields Calculator

The Calculator Stack

The calculator is made up of a stack of registers, each of which can hold:

- Field quantities.
- Functional or constant scalars and vectors.
- Geometries — points, lines, surfaces, or volumes — on which a field quantity is to be evaluated.

To perform a computation on the field solution, you must first load a basic field quantity into a register on the stack. Once a quantity is loaded into a register, it can be:

- Manipulated using mathematical operations such as curls, gradients, cross products, divergences, and dot products.
- Integrated over lines, surfaces, or subvolumes of the solution region — either predefined surfaces, volumes, and lists, or lines, surfaces, and volumes that were defined using the **Draw** commands.
- Plotted on a point, line, surface, or volume.
- Exported to a file, allowing you to superimpose saved solutions.

Related Topics

Registers

Using the Fields Calculator

Registers

Calculator registers hold field quantities, numbers, vectors, and geometries. No registers are created until you load something into the calculator; therefore, this part of the window is initially blank. As items are loaded into the calculator, it creates new registers to hold them.

Each register is labeled with its contents as follows:

Vec	Vector quantities, which have both direction and magnitude at each point in space. The x-, y-, and z-components of these quantities are stored in the register.
Scl	Scalar quantities, which have a magnitude only.
CVc	Complex vector quantities.
CSc	Complex scalar quantities.
Pnt	Points.
Lin	Lines.
Srf	Surfaces.
Vol	Volumes.
SclLin	Scalar value on a line.
VecLine	Vector value on a line.
SclSrf	Scalar value on a surface.
VecSrf	Vector value on a surface.

When examining calculator registers, keep the following in mind:

- To move or delete calculator registers, use the stack commands.
- To save a register to a disk file, use the **Write** command.

Related Topics

Enlarging the Register Display Area

Units of Measure

Enlarging the Register Display Area

If there are too many registers to fit into the display area, do one of the following:

- Use the scroll bars to view the hidden registers.
- Enlarge the calculator window using the window's borders.

Related Topics

Registers

Units of Measure

Unless you are prompted specifically for the unit of measure, all measurements should be assumed to be in SI base units, not model units.

Related Topics

Registers

Stack Commands

Use these commands to manipulate the registers in the calculator stack.

Push

Reloads the quantity in the top register onto the top of the stack, creating a new register. The contents of the top two registers are identical.

Pop

Deletes the top register from the stack.

RlUp

Rolls the top register to the bottom of the stack, moving the other registers up the stack.

RlDn

Rolls the bottom register to the top of the stack, moving the other registers down the stack.

Exch

Exchanges the top two registers in the stack.

Clear

Clears the contents of the stack.

Undo

Use this command to undo the effect of the last operation you performed on the contents of the top register. Successive **Undo** commands act on any previous operations.

Note	You cannot undo a simple operation such as loading a field quantity, constant, function, or geometry into the calculator. Instead, use the **Pop** or **Clear** commands to delete these items from the calculator stack.

Related Topics

Using the Fields Calculator

Input Commands

Use the following commands to load data onto the top of the calculator stack:

Quantity	Basic quantities, such as current, magnetic field, etc.
Geometry	Geometries such as planes, points, polylines, and volumes.
Constant	Predefined constants such as π, ε_0, and conversion factors between various units of measurement.
Number	Vector and scalar constants, including complex numbers.
Function	Previously saved calculator registers containing field quantities.
Geom Settings	Vector and scalar math functions.
Read	Reads an input saved externally.

These quantities can be manipulated using the **Stack** commands, **General** commands, **Scalar** commands, and **Vector** commands. The results of these calculations can then be examined using the **Output** commands.

Related Topics

Using the Fields Calculator

Quantity Command

The **Quantity** command loads a field quantity into the top register of the calculator. The quantities available depend upon the solution and design type. These are the same field quantities available for plotting from the **Maxwell3D>Fields** and **Maxwell2D>Fields** menu.

Related Topics

Plotting Standard Field Quantities

Using the Fields Calculator

Geometry Command

The Fields Calculator **Geometry** command opens a dialog that lets you select a geometry to load into the top register of the calculator. Do this to:

- Find the value of derived field quantities on any point, line, surface, or volume.
- Plot quantities directly from the calculator.

- Display a previously defined isosurface, maximum or minimum field point using the **Draw** command.

The following types of geometries are available:

Point - See drawing a point object. Points you draw are listed in the history tree, and in the Calculator **Geometry** dialog when you select Point.
Line- See drawing a line object. Lines you draw are listed in the history tree, and in the Calculator **Geometry** dialog when you select Line. To set the number of points on a line, see Geom Settings.
Surface - Sheet objects and face lists which you can make, are listed under surface in the history tree and in the Calculator **Geometry** dialog when you select Surface. Due to the ambiguity of the normal vector of a sheet, the result may require a multiplication by (1) or (-1).
Volume - 3D objects, Regions, and object lists of 3D objects including AllObjects are available in the Calculator **Geometry** dialog when you select Volume.
Coord -Coordinate systems are available in the Calculator **Geometry** dialog when you select Coord.

To load a geometry into the calculator:

1. In the Fields Calculator, click **Geometry**.

   The **Geometry** dialog box appears.
2. Select a geometry type.

   A list of all available geometries appears.

Note	• Non-convex 3D non-model solids are not supported for Field Overlay Plots and Fields Calculator computations. • Object lists containing non-model object(s) are supported for Field Overlay Plots but are not supported for Fields Calculator computations.

3. Click the geometry.
4. Click **OK** to load the geometry.

Note	Consider a box (Box2) that is completely enclosed in a bigger box (Box1), so that no faces of Box2 are touching any faces of Box1. If you explicitly subtract Box2 from Box1, any calculation on the surface (faces) of Box1 will use the 6 exterior faces and the 6 interior faces. Any calculation on the volume of Box1 will use the difference in volume between Box1 and Box2. If you do not explicitly subtract Box2 from Box1, the inner box is only implicitly subtracted. Any calculation on the surface of Box1 in this case will use only the 6 exterior faces of Box1. Any calculation on the volume of Box1 will use the entire volume without subtracting the volume of Box2.

Related Topics

Using the Fields Calculator
Input Commands
Domain command
Export Command
Geom Settings

Constant Command

The **Constant** command loads one of these predefined constants or a conversion constant into the top register of the calculator:

Pi	$\pi = 3.14159265358979$
Epsi0	The permittivity of free space, $\varepsilon_0 = 8.85418782 \times 10^{-12}$ F/m.
Mu0	The permeability of free space, $\mu_0 = 4\pi \times 10^{-7}$ H/m.
C	The speed of light in vacuum, $c = 2.99792458 \times 10^{8}$ m/s.
conversion constant	Displays the **Enter Units Conversion Factor** dialog box. This dialog box lists a range of **Quantities** (such as frequency, resistance, and others) along with a list of Units (Hz to Thz, and rps) to convert **From** and **To**. The ratio of the **Units from** to the **Units to** is displayed for the selected values as the **Conversion Factor**.

Related Topics

Using the Fields Calculator
Input Commands

Number Command

The **Number** command enters one of the following into the top register of the calculator:

Scalar	A scalar constant. To enter a constant scalar number: 1. Click **Number**. The **Input Number** dialog box appears. 2. Select **Scalar**. 3. Type the scalar value in the **Value** box. 4. Click **OK** to load the number into the top register.
Vector	A vector constant.

	To enter a constant vector: 1. Click **Number**. The **Input Number** dialog box appears. 2. Select **Vector**. 3. Enter the x-, y-, and z-components of the vector. 4. Click **OK** to load the vector into the top register.
Complex	A complex constant. Complex constants are entered in the form *C=A+jB*, where *A* represents the real part of the constant and *B* represents the imaginary part. 1. Click **Number**. The **Input Number** dialog box appears. 2. Select **Scalar** or **Vector**. 3. Select **Complex**. 4. Enter the real and imaginary components of the number. 5. Click **OK** to load the number into the top register.

Related Topics

Using the Fields Calculator
Input Commands

Function Command

Any functions you use must be defined before using this operation.
Enters one of the following into the top register of the calculator:

Scalar	A scalar function. To enter a function: 1. Click **Function**. The **Function** dialog box appears. 2. Select **Scalar**. 3. Select the function from the list. 4. Click **OK** to load the functional scalar into the top register.
Vector	A vector function, in which the values of the vector's x-, y-, and z-components are given by functions. To enter a functional vector: 1. Click **Function**. The **Function** dialog box appears. 2. Select **Vector**.

	3. Select the function from the list. 4. For each component of the vector, click **SetX**, **SetY**, and **SetZ**. 5. Click **OK** to load the functional vector into the top register.

Note	• The predefined variables **X**, **Y**, **Z**, **RHO**, **THETA**, **R**, and **PHI** and any functions that you created can be used to define functional scalar and vector quantities. • Use of the Global Coordinate System is assumed. Local coordinate systems are not used.

Related Topics

Using the Fields Calculator

Input Commands

Geometry Settings

Clicking the **Geom Settings** button opens the **Geometric Settings** dialog box. The dialog box allows you to specify the line discretization, the number of equally-spaced points used to integrate fields and other quantities on a line. The default is 1000 points.

To set the line spacing for geometry settings:

1. In the Fields Calculator, click **Geom Settings**.

   The **Geometry Settings** dialog box appears.
2. Enter a value in the **Line Discretization** box, and click **OK**.

Related Topics

Using the Fields Calculator

Input Commands

Reading an External Input

This command copies the contents of a disk file into the top register. The register must be one that has been saved using the Write output command.

To read an external input into the register:

1. Click **Read**.

   The **Open** dialog box appears.
2. Use the file browser to specify the register's file name and directory path. A .reg extension is automatically assumed for register files, and click **Open**.

   The contents of the file are copied to the top register in the stack.

Related Topics

Using the Fields Calculator

Input Commands

General Commands

Use these Fields Calculator commands to perform operations on both vector and scalar quantities.

+ (Add)	/ (Divide)	Smooth
- (Subtract)	Neg	Complex
* (Multiply)	Abs	Domain

+ (Add)

Adds the quantities in the top two registers of the calculator.

– (Subtract)

Subtracts the quantity in the top register from the quantity in the second register. The two registers must hold the same type of quantity (both scalar or both vector). You cannot subtract a scalar from a vector (or vice versa).

*** (Multiply)**

Multiplies the quantity in the top register by the quantity in the second register. One of the two registers must contain a scalar value; the other register can be either a scalar or a vector.

/ (Divide)

Divides the quantity in the second register by the quantity in the top register. The second register must contain a scalar value; the top register can be either a scalar or a vector.

Neg

Changes the sign of the quantity in the top register.

Abs

Takes the absolute value of the quantity in the top register.

Smooth

Smooths the quantity in the top register. Because of the numerical solution technique used, field values are not always continuous across the boundaries of the individual elements that

make up the finite-element mesh. Smoothing makes the values continuous. In general, use smoothing before plotting a quantity.

Complex

These commands perform operations on a complex quantity in the top register. Complex quantities are indicated by a **C** at the beginning of the register label. They can be represented in terms of real and imaginary components, or in terms of magnitude and phase:

$$C = A + jB = Me^{j\phi}$$

where:

- *A* is the real part of the complex number.
- *B* is the imaginary part of the complex number.
- *M* is its magnitude, which is equal to sqrt(A^2 + B^2).
- ϕ is its phase, which is equal to **atan**(B/A).

The **Complex** commands let you do the following:

Real	Takes the real part of the complex quantity (A).
Imag	Takes the imaginary part of the complex quantity (B).
CmplxMag	Takes the magnitude of the complex quantity (M). Due to interpolation issues, the sequence of calculations may cause a loss of accuracy. It is best to define the points, separately obtain the value of the real part, then the imaginary part, and use those values to calculate the magnitude and phase. For the sequence for using the Fields Calculator to obtain the real and imaginary parts, see the procedure here.
CmplxPhase	Takes the phase of the complex quantity (ϕ).
Conj	Takes the complex conjugate of the quantity in the top register. If a complex number is given by $C = A + jB$, its complex conjugate is given by $C^* = A - jB$.
AtPhase	Specify the phase angle, ωt, at which a field quantity is evaluated. These quantities can be represented in the form: $$A(x,y,z,t) = A(x,y,z)\cos(\omega t + \theta(x,y,z))$$ where • ω is the angular frequency at which the quantities are oscillating, specified during the solution.

	• θ(x,y,z) is the phase angle (the offset from a cosine wave that peaks at t=0). Entering the phase angle lets you compute the real part of the field's magnitude at different points in its cycle.
CmplxReal	Converts the real scalar of the top register to the real part of a complex number.
CmplxImag	Converts the real scalar of the top register to the imaginary part of a complex number.
ComplxPeak	Calculates the peak value of a given complex vector. Intuitively, this calculates the maximum magnitude of the equivalent real vector in a waveform.

Domain

This limits a calculation to the volume you specify. The domain filter works for scalars, vectors, complex scalars and complex vectors. This operation requires the top two entries of the stack to be a volume geometry and a numeric field quantity. To do this:

1. Load the field quantity into the top register, and perform any necessary operations on it.
2. Load the volume using the **Geometry** command.
3. Click **Domain**.

   The **Domain** command is often used to limit a calculation or plot to the intersection of a surface and an object or group of objects. If you export a domain filtered numeric, points that are filtered out by the domain will not be written out.

Related Topics

Using the Fields Calculator

Export Command

Steps for Calculating the Complex Vector Electric Field

Steps for Calculating the Complex Vector Electric Field

These are the field calculator steps to obtain the real part, the imaginary part, and the magnitude of the x-directed, y-directed, and z-directed components of the phasor electric field. For each of these vector components, the magnitude should be equal to sqrt(real^2+imag^2), but the need to interpolate values and the calculation sequence means that Maxwell does not give this value unless the specified location is directly on a mesh element node.

1. Calculate real part of complex vector electric field (in x, y, and z directions):
   a. Qty > E
   b. Complex > Real
   c. Geometry > Point > fieldcalc_point
   d. Value
   e. Eval
2. Calculate imaginary part of complex vector electric field (in x, y, and z directions):

a. Qty > E
b. Complex > Imag
c. Geometry > Point > fieldcalc_point
d. Value
e. Eval

Use the real and imaginary components to manually calculate the magnitude as the sqrt ($Real^2+imag^2$).

Scalar Commands

Use these commands to perform operations on scalar quantities.

Vec?	Makes the scalar quantity in the top register a vector component.
1/x	Takes the inverse of the scalar quantity in the top register.
Pow	Raises a scalar quantity to the power you specify.
Square Root	Takes the square root of the quantity in the top register.
Trig	Takes a trigonometric value of the value in the top register of the calculator stack.
d/d?	Takes the partial derivative of the quantity in the top register with respect to x, y, or z as chosen in the pull-down menu.
Integral	Takes the integral of a scalar quantity over a volume, surface, or line.
Min	Computes the minimum of a scalar field quantity on a line, surface, or volume.
Max	Computes the maximum of a scalar field quantity on a line, surface, or volume.
Gradient	Computes the minimum of a scalar field quantity on a line, surface, or volume.
Ln	Calculates the log (base e) value.
Log	Calculates the logarithmic value (base 10).
Mean	Takes the mean of the quantity in the top register.
Std	Takes the standard deviation of the quantity in the top register.

Related Topics

Using the Fields Calculator

Vec? Command

Makes the scalar quantity in the top register a vector component. Choose from the following:

VecX	The x-component of a vector.

VecY	The y-component of a vector.
VecZ	The z-component of a vector.

Related Topics

Using the Fields Calculator
Scalar Commands

1/x (Inverse) Command

Takes the inverse of the scalar quantity in the top register.

Related Topics

Using the Fields Calculator
Scalar Commands

Pow Command

Raises a scalar quantity to the power you specify.

To raise a scalar quantity to a power:

1. Enter the quantity into the calculator.
2. Enter the exponent to which it is to be raised into the calculator.
3. Click **Pow**.
   The results are displayed in the top register.

Related Topics

Using the Fields Calculator
Scalar Commands

(Square Root) Command

Takes the square root of the quantity in the top register.

Related Topics

Using the Fields Calculator
Scalar Commands

Trig

Takes one of the following trigonometric values of the value(s) in the top register(s) of the calculator stack:

Sin	Sine.
Cos	Cosine.
Tan	Tangent.
ASin	Arcsine.
ACos	Arccosine.
ATan	Arctangent.
ATan2	A two-argument version of the **ATan** function. Takes the y and x coordinates of the point (in the top two stack registers).

Related Topics

Using the Fields Calculator
Scalar Commands

d/d? (Partial Derivative) Command

Takes the partial derivative of the quantity in the top register:

d/dx	Takes the partial derivative of the quantity with respect to x.
d/dy	Takes the partial derivative of the quantity with respect to y.
d/dz	Takes the partial derivative of the quantity with respect to z.

Related Topics

Using the Fields Calculator
Scalar Commands

d/d? (Partial Derivative) Command

Takes the partial derivative of the quantity in the top register:

d/dx	Takes the partial derivative of the quantity with respect to x.
d/dy	Takes the partial derivative of the quantity with respect to y.
d/dz	Takes the partial derivative of the quantity with respect to z.

Related Topics

Using the Fields Calculator

Scalar Commands

∫

(Integral) Command

Takes the integral of a scalar quantity over a volume, surface, or line. The top register must contain a geometry, and the second register must contain the scalar quantity to be integrated.

To perform an integration:

1. Load a quantity into the top register of the calculator, and perform any required operations on it.
2. Use one of the **Geometry** commands to load the line, surface, or volume over which the quantity is to be integrated.

Note	If you computed the tangent or normal of the quantity to be integrated, you do not have to load a geometry onto the calculator stack. Maxwell integrates the tangential or normal component of the quantity over the line on which you computed its tangent, or the surface on which you computed its normal.

3. Click the **Integral** command to integrate the scalar quantity over the geometry. In 2D designs, you must further select either **XY** or **RZ**.

To find the numerical results of an integration, use the **Eval** command.

Note	In **Maxwell2D**, the Fields Calculator accounts for the difference between XY and RZ models when handling data and results. The integral command will perform an XY integral on the model assuming a 1 meter depth, while for an RZ model it will perform an integral on the 360 degree equivalent volume.

Related Topics

Using the Fields Calculator

Scalar Commands

Min Command

Computes the minimum of a scalar field quantity on a line, surface, or volume. Two options are available:

Value	Finds the magnitude of the minimum value of the field.
Position	Finds the point where the minimum field value occurs. You can then: • Plot the minimum field value at the point using the **Plot** command. • Plot basic field quantities at the point. • Load the point into the calculator. • Change the point's location.

These commands operate in the same way as the **Max** commands. Use the **Eval** command to display the actual minimum field value or the coordinates of the point where it occurs.

Related Topics

Using the Fields Calculator

Scalar Commands

Max Command

Computes the maximum of a scalar field quantity on a line, surface, or volume. Two options are available:

Value	Finds the magnitude of the maximum value of the field.
Position	Finds the point where the maximum field value occurs. You can then: • Plot the maximum field at the point using the **Plot** command. • Plot field quantities at the point. • Load the point into the calculator. • Change the point's location.

To compute the maximum field value:

1. Load a field quantity into the calculator, and perform any necessary operations on it. Keep the following in mind:
   - You cannot find the maximum value of a vector quantity. Therefore, make sure that the result is a scalar.
   - Before computing the maximum value of a complex quantity, you must convert it to a scalar quantity using the**Cmplex/Real** or **Cmplex/AtPhase** commands.
2. Load a point, line, or volume into the calculator using one of the **Geometry** commands.
3. Do one of the following:
   - Click **Max/Value** to compute the maximum field value on the geometry.
   - Click **Max/Position** to identify the point at which this value occurs.

Use the **Eval** command to display the actual maximum field value or the coordinates of the point where it occurs.

Related Topics

Using the Fields Calculator
Scalar Commands

(Gradient) Command

Computes the gradient of the scalar quantity in the top register.

Related Topics

Using the Fields Calculator
Scalar Commands

Ln Command

Takes the natural logarithm (base e) of the scalar quantity in the top register.

Related Topics

Using the Fields Calculator
Scalar Commands

Log Command

Takes the logarithm (base 10) of the scalar quantity in the top register.

Related Topics

Using the Fields Calculator
Scalar Commands

Mean Command

Takes the mean of the quantity in the top register.

Related Topics

Using the Fields Calculator
Scalar Commands

Std Command

Takes the standard deviation of the quantity in the top register.

Related Topics

Using the Fields Calculator
Scalar Commands

Vector Commands

Use these commands to perform operations on vector quantities.

Scal?	Replaces the vector in the top register with a scalar quantity that is a component of the vector.
Matl	Multiplies or divides the vector field quantity in the top register by a material property.
Mag	Takes the magnitude of the vector quantity in the top register. The magnitude of a complex vector is defined to be the length of the real vector resulting from taking the modulus of each component of the original complex vector.
Dot	Takes the dot product of the vector quantities in the top two registers
Cross	Takes the cross product of the vector quantities in the top two registers.
Divg	Takes the divergence of the vector quantity in the top register.
Curl	Takes the curl of the vector quantity in the top register.
Tangent	Computes the tangential component of a vector quantity along a line.
Normal	Computes the normal component of a vector quantity on a surface such as a

	cutplane or object surface.
Unit Vec	Computes the normal or tangent unit vector. The unit vector is a "wild card" entry. The context is specified at the time of plotting, integrating, or report generation.
XForm	Transforms the vector quantity in the top register from Cartesian coordinates into either Spherical or Cylindrical coordinates

Scal? Command

Replaces the vector in the top register with a scalar quantity whose value is a component of the vector. Choose from the following:

ScalarX	Returns the x-component of the vector.
ScalarY	Returns the y-component of the vector.
ScalarZ	Returns the z-component of the vector.

Related Topics

Using the Fields Calculator

Vector Commands

Matl Command

Multiplies or divides the vector field quantity in the top register by a material property. At each tetrahedron, the field quantity is multiplied or divided by the value of the selected material property — taking the different material attributes of each object into account.

To multiply or divide a vector quantity by a material property:

1. Click **Matl**.

   The **Material Operation** dialog box appears.
2. Select a property from the **Material Properties** list. Options include the following, depending on what solution type is assigned:
   - Permeability (mu): defined as $\mu_0\mu_r$

     where μ_0 is the permeability of free space
   - Differential Permeability (dmu)
   - Conductivity (cond)
   - Permittivity (epsi)

- Mass Density: based on the value of the Mass Density material property. MassDensity is treated like a named expression. Selecting MassDensity disables the Operation radio buttons for Multiply or Divide in the Material Operations dialog.
- Admittivity

3. Select an operation — **Multiply** or **Divide**.
4. Click **OK** to multiply or divide the field quantity by a material property or **Cancel** to stop the operation. If you selected MassDensity and click **OK**, a scalar named expression MassDensity is pushed onto the stack.

Related Topics

Using the Fields Calculator
Vector Commands

Mag Command

Takes the magnitude of the vector quantity in the top register. The magnitude of a complex vector is defined to be the length of the real vector resulting from taking the modulus of each component of the original complex vector.

With a complex vector on the calculator stack, the **Mag** button returns a nonnegative scalar. In previous software versions, this command returned a complex scalar.

Related Topics

Using the Fields Calculator
Vector Commands

Dot Command

Takes the dot product of the vector quantities in the top two registers.

Related Topics

Using the Fields Calculator
Vector Commands

Cross Command

Takes the cross product of the vector quantities in the top two registers.

Related Topics

Using the Fields Calculator

Vector Commands

Divg Command

Takes the divergence of the vector quantity in the top register.

Related Topics

Using the Fields Calculator
Vector Commands

Curl Command

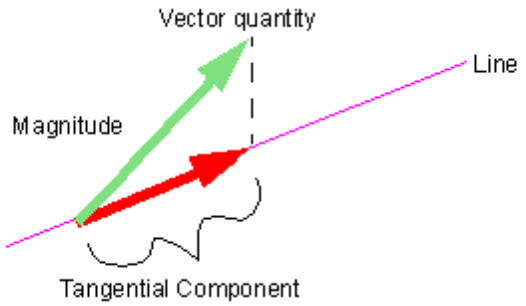
Takes the curl of the vector quantity in the top register.

Related Topics

Using the Fields Calculator
Vector Commands

Tangent Command

Computes the tangential component of a vector quantity along a line.



To take the tangent of a vector:

1. Load a vector quantity into the top register.
2. Load a line into the top register using the **Geometry/Line** command.
3. Click **Tangent**.

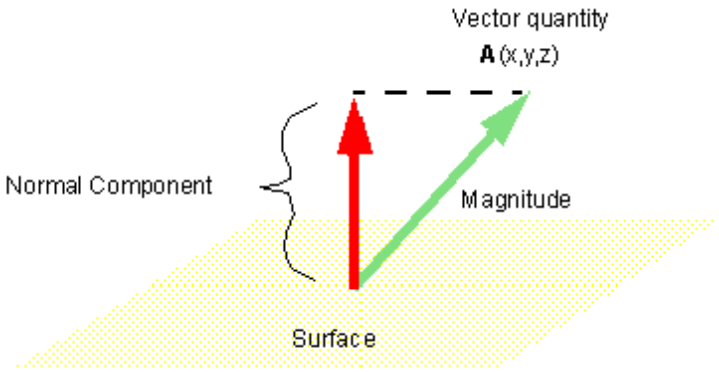
Related Topics

Using the Fields Calculator
Vector Commands

Normal Command

Computes the normal component of a vector quantity on a surface such as a cutplane or object surface. This is the equivalent of taking the dot product of the quantity with the surface's unit normal vector:

$$Normal = \boldsymbol{A}(x,y,z) \bullet \hat{\boldsymbol{n}}$$



To take the normal of a vector:

1. Load a vector quantity into the top register.
2. Load a surface into the top register using the **Geometry/Surface** command.
3. Click **Normal**.

Note	Because surface normals of sheets are not well defined the fields calculator can produce incorrect results if an expression is evaluated on a sheet. To enforce the correct direction of the surface normal of a sheet, a faceted 3D object (such as a box) can be defined such that one of its planar faces is coincident with the sheet. Because surface normals of a valid object are always defined in an outward direction in Maxwell, the fields calculator uses the surface normal of the face of the 3D object that is coincident with the sheet.

Related Topics

Using the Fields Calculator

Vector Commands

Unit Vec Command

Computes the normal or tangent unit vector. The unit vector is a "wild card" entry. The context is specified at the time of plotting, integrating, or report generation.

Select from the following:

Tangent	Computes the unit vector tangent to the line in the top register of the calculator stack.
Normal	Computes the unit vector normal to the surface in the top register of the calculator stack.
CoordSys (X)	Computes the unit vector in the X-dimension of the relative coordinate system in the top register of the calculator stack. Add the relative CS as a geometric object using the **Geometry/Coord** command.
CoordSys (Y)	Computes the unit vector in the Y-dimension of the relative coordinate system in the top register of the calculator stack. Add the relative CS as a geometric object using the **Geometry/Coord** command.
CoordSys (Z)	Computes the unit vector in the Z-dimension of the relative coordinate system in the top register of the calculator stack. Add the relative CS as a geometric object using the **Geometry/Coord** command.

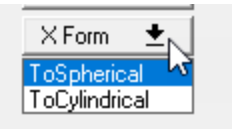
Related Topics

Using the Fields Calculator

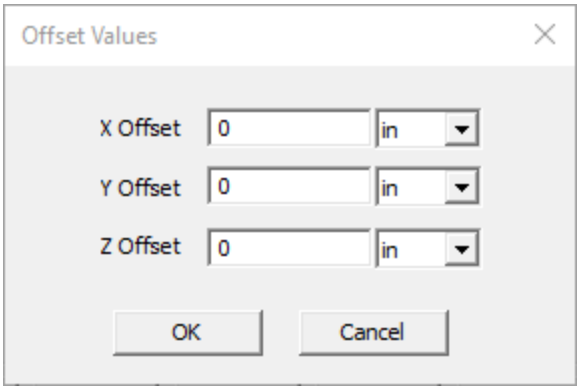
Vector Commands

XForm Command

Fields are defined in the Cartesian coordinate system. In many applications it is much more convenient to use a cylindrical or spherical coordinate system – for example, when the object is of cylindrical or spherical shape. The **XForm** command allows the user to transform vectors into either the cylindrical or spherical coordinate system. Selecting **XForm** displays a selection menu for the coordinate system (**ToCylindrical** or **ToSpherical**):



After you select the coordinate system you want, the **Offset Values** dialog displays.



Here you enter X, Y, and Z offsets to be used as the origin of the new coordinate system.

Click **OK** to compute the offset coordinate transformation to apply to the (complex) vector quantity in the top register on the calculator stack based on your selection of Cylindrical or Spherical coordinates and your input of X, Y, and Z offsets. The new expression will be pushed onto the stack. This provides the origin of the new coordinate system. It will be something like CVc : ToCylindrical(<Ex,Ey,Ez>,offset<0mil,0mil,0mil>) or CVc : ToSpherical (<Ex,Ey,Ez>,offset<0mil,0mil,0mil>) respectively.

After you transform the quantities, you can then access the vector components in the Fields Calculator.

You will then use the ScalarX/ScalarY/ScalarZ commands under **Scal?**. For a vector converted to cylindrical coordinates, ScalarX gives the radial component, ScalarY gives the phi (polar) component, and ScalarZ gives the Z component. For a vector in spherical coordinates, ScalarX gives the radial component, ScalarY gives the phi (polar) component and ScalarZ gives the theta (azimuthal) component.

Related Topics

Using the Fields Calculator

Vector Commands

Output Commands

Use these commands to compute or evaluate expressions and to output the data in the calculator.

Value command	Computes the value of a field quantity at a point.
Eval command	Numerically evaluates and displays the results of calculator operations.
Write command	Saves the contents of the top register to a disk file.

Export command	Saves field quantities in a format that can be read by other modeling or post-processing software packages.

Related Topics

Using the Fields Calculator

Value Command

This computes the value of a field quantity at a point. Use it to find:

- The magnitude of a scalar field quantity at that point.
- The x-, y-, and z-components of a vector field quantity at that point.

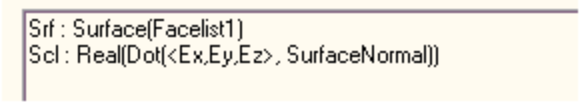
To find the value of a field quantity at a point:

1. Load the field quantity into the top register, and perform any needed operations on it.
2. Load the appropriate point into the calculator using the **Geometry/Point** command.
3. Click **Value**.

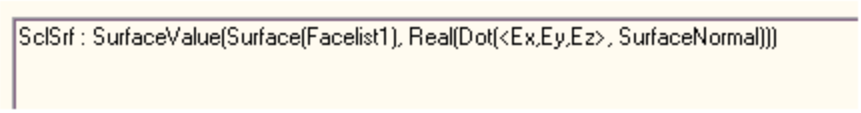
To view the numerical results of this operation, use the **Eval** command.

You can also use the **Value** command to access the intermediate SurfaceValue function

For example, after inputting an expression for a quantity, such as an E field, and then selecting a surface geometry, the calculator stack displays something like this.



Clicking the **Value** command changes the display to the following, showing the intermediate SurfaceValue function.



In this case SurfaceValue provides the x, y, z, coordinates of the FEM mesh and Lagrangian points so you can use **Write** to generate an .fld file containing an evaluated scalar quantity at those points. In general for **Value**:

1. Enter any quantity onto the stack.
2. Enter a volume / surface / line / point onto the stack.
3. Press the **Value** button and you will get an appropriate geometry value on the stack.

Now you can perform suitable operations such as Write, Integrate, etc. For PointValue you can also do **Eval**.

Related Topics

Using the Fields Calculator

Eval Command

This command numerically evaluates and displays the results of calculator operations such as integrations, maximum or minimum field computations, field values at points, and so forth. The quantity to be evaluated must be in the top register. The **Eval** command computes the numerical results of the operation, which replace the contents of the register.

For instance, to find the current around a loop, you must numerically evaluate the following

$$I = \oint H \bullet dl$$

integral for that loop: .

Since **H** and **I** are complex quantities, you first need to evaluate the real part of **H** to obtain the real part of **I**, then evaluate the imaginary part of **H** to obtain the imaginary part of **I**. To do this:

1. Load **H** into the calculator using the **Qty** command.
2. Take the real part of **H** using the **Cmplex/Real** command.
3. Load the rectangular loop using the **Geom/Line** command. Create the loop, a closed polyline, to integrate over.
4. Click **Tangent** to get the component of **H** along the line.
5. Take the integral around the loop using the $\int$ command.
6. Click **Eval** to evaluate the integral. The real part of **I** appears in the top register.
7. Repeat this process using the imaginary part of **H** (found with the **Complex/Imag** command) to obtain the imaginary part of **I**.

Related Topics

Using the Fields Calculator

Write Command

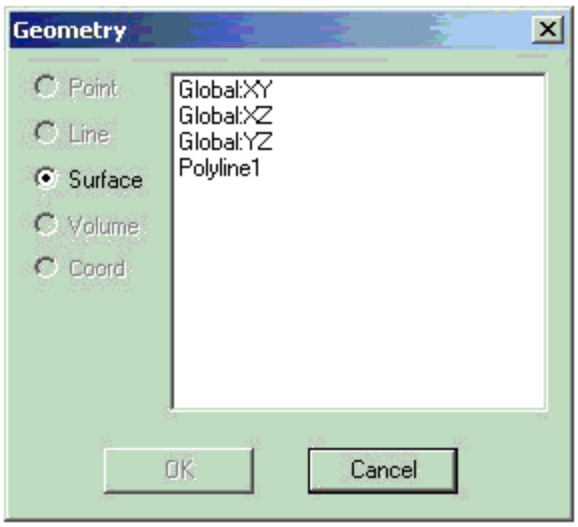
This command saves the contents of the top register to a disk file. Use this command to:

- Save registers for use during a later post-processing session.
- Save a field quantity for use when post processing a different model.

Note	Use of the Global Coordinate System is assumed. Local coordinate systems are not used.

To save a register:

1. Click **Write**.
2. If the register includes numeric with a constrained quantity (such as jsurf), you see a dialog that gives a choice of constraining geometries. For example:



3. Select the geometry of interest, and select **OK**.

   This displays a file browser.
4. Use the file browser to specify the register's file name and directory path. A .reg extension is automatically assigned to register files and a .fld extension is assigned to field files. You can choose to save both .reg and .fld files, or either one.
5. Click **OK**.

   The contents of the file are saved to the file you specified.

Related Topics

Using the Fields Calculator

Export Command

This command opens the **Export Solution** dialog, from which you can export the field quantity in the top register to a file, mapping it to a grid of points. Use this command to save field quantities in

a format that can be read by other modeling or post-processing software packages. Two options are available for defining the grid points on which to export:

Input grid points from file	Maps the field quantity to a customized grid of points. Before using this command, you must create a file containing the points and units.
Calculate grid points	Maps the field quantity to a three-dimensional cartesian grid. You specify the dimensions and spacing of the grid in the x, y, and z directions, with units that you specify. The initial units are taken from the model.

Note	Use of the Global Coordinate System is assumed. Local coordinate systems are not used.

To export a field quantity to a customized grid:

1. Load the quantity into the top register for the fields calculator, and perform any operations on it.
2. If desired, load a volume using the **Geometry** command.

   You can use the Domain command to limit the calculation to the volume you specify. If you export a Domain filtered numeric, points that are filtered out by the domain will not be written out.
3. Click the **Export button in the Fields Calculator.**

   This opens the **Export Solution** dialog.
4. Type or select the name of the file in which the field quantity is to be saved in the **Output File Name** text box. You can use the file icon to open the file browser to specify the file name and directory path. A **.reg** extension is automatically assigned to this file.
5. Click either the **Input grid points from file** button if you have a created a **.pts** file containing the grid points, or click the **Calculate grid points** button.
   - If you select **Input grid points from file**, either type the name and directory of the file containing the points on which the field is to be mapped, or, click on the file icon and use the file browser to locate the point file (**.pts** extension).

Note	By default, the field calculator uses **SI** units. Optionally, you can also specify the units that you want while exporting. Add a line in the beginning of the .pts file to specify the units. For example, if you want to use **mm**, a sample file would like this:
	
	`Unit=mm`
	`-5.5 -5.5 -5.21475`
	`-5.5 -5.5 -5.14425`
	`-5.5 -5.5 -5.07375`
	`-5.5 -5.5 -5.021`

   - If you select **Calculate grid points** button, you can specify the coordinate system as Cartesian, Cylindrical, or Spherical.

Cartesian: for each grid dimension on X, Y, and Z, enter the Minimum, Maximum, and grid point spacing.

Cylindrical: for each dimension Rho, Phi, and Z, enter the Minimum, Maximum, and grid point Spacing. You can also specify an origin of Offset.

Spherical: for each dimension R, Theta, and Phi, enter the Minimum, Maximum, and grid point Spacing. You can also specify an origin of Offset.

Note	When you export fields on a 1D or 2D line/surface from the field calculator, the start and stop values must be the same for one or two of the coordinate system start/stop ranges. If you specify a zero spacing for a dimension, the export uses only the minimum value.

The default coordinate system is Cartesian. The default offset will be all zeros. The length units will default to model unit and default angle unit will be degree. At the start the minimum/maximum/Spacing entries are blank. The user-entered values are not remembered when the dialog is closed.

6. For larger files, you may want to uncheck the **Include points in output file** box. If you uncheck the box, the file header will include minimum, maximum and spacing information from which you can recalculate the grid points.
7. Click **OK** to export the file.

   The field quantity is mapped to the grid and saved to the file you specified (**.reg** extension.).

Related Topics

Using the Fields Calculator

Input Commands

Domain command

Calculating Derived Field Quantities

The **Named Expressions** panel displays expressions that can be included in register definitions by name. You can add additional expressions to the Named expression list by creating the expression in the register display area, and the clicking the Add button. This lets you add to the Named expression library.

When a Maxwell design is open and a Solution Setup has been performed, numerous predefined named expressions may be available depending upon the solution type. Generally, predefined expressions related to electric or magnetic field magnitude, current magnitude, vector field values, energy, and losses are available

Click on a named expression to select it. When a named expression has been selected, the **Copy to Stack** button is activated. Click **Copy to Stack** to push the expression on the top of the stack.

Related Topics

Named Expressions

Named Expression Library

Named Expression Library

The named expression library in the Fields Calculator provides a way to conveniently calculate frequently used quantities. The library comes with several predefined expressions. You can combine calculator Input commands in any legal fashion, including complex quantities, to produce new named expressions.

- Adding named expressions to the Fields Calculator expression library
- Copying named expressions to the Calculator Stack
- Saving named expressions to a Personal Library
- Loading named expressions
- Deleting named expressions that you added

To add a named expression of your own to the Fields Calculator list:

1. In the register display area, create the expression by using the calculator Input commands.

   You can combine input commands in any legal fashion, including the use of complex quantities. If you select an input command that is not legal for a current operation, you receive an error message.
2. When you finish creating the expression, click **Add in the Named Expressions panel**.

   The **Named Expression** dialog box appears.
3. Type a name for the expression in the **Name** text box.

   The new expression is added to the list of named expressions.

To copy a named expressions to the Calculator Stack

- You can scroll through the list, select any desired named expression, and click **Copy to Stack** to move it to the calculator stack, where you can use it to generate calculated outputs.

To delete named expressions that you added:

When the **Named Expression** list contains one or more user-defined expressions, the **Delete** and **Clear All** buttons are active (you cannot delete or clear the predefined named expressions.)

- To delete the selected user-defined named expression, click **Delete**. To delete all user-defined named expressions, click **Clear All**.

To save one or more named expressions for the Fields Calculator to a personal Library:

1. Click the **Save To** button on the Fields Calculator.

   The **Select Expressions for Saving** dialog displays.
2. If any new named expressions exist, you can select one or more to save to a file.
3. Give a file name, and click **OK** to save the file.

To load named expressions for the Fields Calculator from a personal library:

1. From the Fields Calculator, click **Load From**.

   This displays a file browser that you can use to search for existing .clc files.
2. Select the library to load and click **OK**.

This loads the expression file you have selected.

Related Topics

Calculating Derived Field Quantities

Exiting the Fields Calculator

Click **Done** to exit the Fields Calculator.

Related Topics

Using the Fields Calculator

Plotting the Mesh

Before or after the solution is complete, you can plot the finite element mesh on surfaces or within 3D objects.

1. Select a surface or object to create the mesh plot on or within.
   If it does not exist, create it.
2. Click **Maxwell3D>Fields>Plot Mesh** or **Maxwell2D>Fields>Plot Mesh**.
   The **Create Mesh Plot** dialog box appears.
3. Enter a name, or accept the default name.
4. Select the solution to plot from the **Solution** pull-down list.
5. Select the field quantity to plot from the **Field Type** pull-down list.
6. Click **Done**.
   The mesh appears on the surface or object you selected. An icon for the mesh also appears in the **Project** tree under **Field Overlays -- Mesh Plots**.
   If a solution is ongoing, you can select the Mesh Plots icon in the in the Project tree, right-click to display the shortcut menu, and click **Update Plots**. This updates the mesh plot to latest data available. After the last adaptive pass, the Mesh plot is automatically updated.
   If a mesh includes seeding, these effects appear at the start of the adaptive passes. Any mesh adaptation at the start of a sub sequent pass is not plotted until after that pass is completed. This delay ensures that mesh plots and actual solutions remain consistent with each other.
   You can modify an existing plot by selecting the plot and changing the properties. If you modify the geometry of an object, the solution is invalidated, and the mesh plot is removed.
   If you have plotted the mesh of a Solve-Inside object, you can view the internal mesh by using either the clip-plane feature or the Model Analysis dialog box. Also see the discussion here, under **Visualization for Skin Depth Refinement Mesh Results**. The same techniques for visualization for internal mesh plots apply even if seeding is not used.

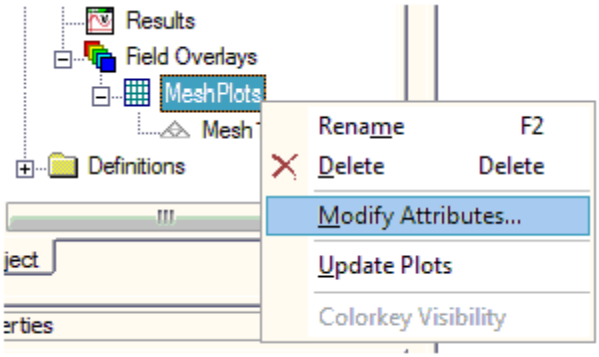
Related Topics

Setting Mesh Plot Attributes

Mesh Feedback Tab

Setting Mesh Plot Attributes

1. Select the Mesh Plots icon under Field Overlays in the Project tree, right-click, and select Modify Attributes to display the **Mesh Plots** dialog.



You can also click **Maxwell 2D** or **Maxwell 3D>Fields>Modify Plot Attributes** , after which the **Select Folder** window appears. Select the folder containing the mesh plot you want to modify, and then click **OK** to display the **Mesh Plots** dialog.

For Mesh plots, the following attributes can be modified:

Plot	A drop down list of available plots.
Scale Factor	The size at which the tetrahedra are displayed. Scaling may let you analyze particular situations better. For example, a scale factor of 80% draws the tetrahedra at 80% of their original size. Use the **Scale factor** slider to increase (move to the right) or decrease (move to the left) the percentage of the tetrahedra size.
Transparency	The degree of transparency for the tetrahedra. This is useful for viewing objects or plots behind the current plot. Use the **Transparency** slider to increase (move to the right) or decrease (move to the left) the transparency of the plot.
Mesh type	Whether to display the tetrahedra as wire frame or shaded, and whether to

	Add Grid.
Mesh Color for Line and Fill	The color for the tetrahedra edge lines and fill. Clicking the button for each displays a color selection dialog.
Surface Only	Whether to plot the surface only, or all tetrahedra inside selected objects.
Real Time	Whether to show changes to a mesh in real time.

2. Click the **Save as default** button if you want the tab's settings to apply to mesh plots created after this point.
3. Select **Real time mode** if you want the changes to take effect immediately in the view window.

   If this option is cleared, click **Apply** when you want to see the changes.
4. Click **Close** to dismiss the dialog box.

Related Topics

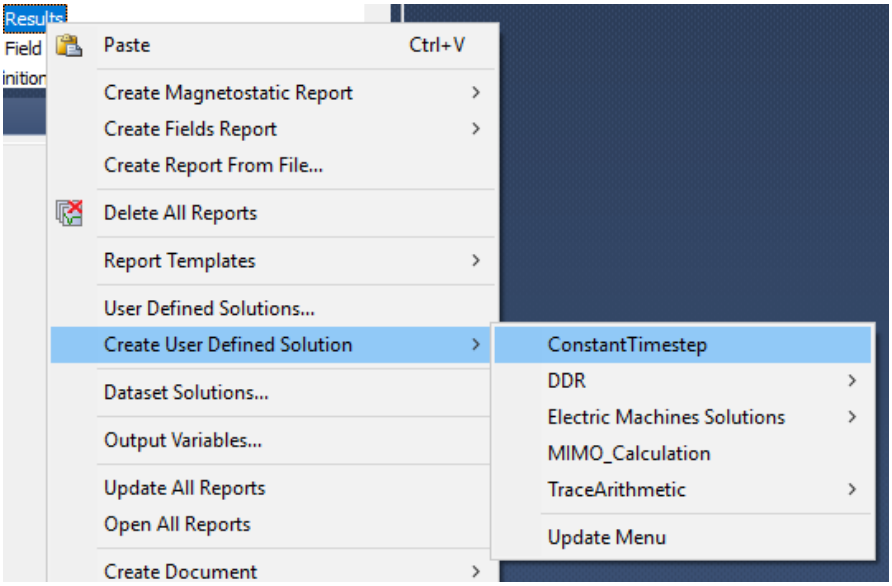
Plotting the Mesh
Plotting Field Overlays

User Defined Outputs: Introduction

User defined outputs (UDOs) allow users to define calculations through IronPython scripts or any .NET language (and used by the IronPython script). The UDO scripts need to be in the **UserDefinedOutputs** directory under either of **syslib**, **userlib** or **Personallib** with any directory structure needed for organization. (The **Lib** directory name is special and its purpose will be explained in a subsequent section.)

The UDO scripts that are in **syslib/UserDefinedOutputs**, **userlib/UserDefinedOutputs**, or **Personallib/UserDefinedOutputs** become available to the user to create "User Defined Solutions" through the **Results>Create User Defined Solution** menu.

Use **Results>Create User Defined Solution>Update Menu** to refresh the menu to include the new UDO scripts that might have been copied to syslib, userlib or Personallib, or exclude them if they have been deleted, after the launch of desktop. Once the user-defined-solution is created, the solution and the calculations defined by UDO become available in Reporter as any other quantities in a new "User Defined" report type. (**Electric Machines Solutions** sub-menu is available only on Windows.)



Named Probes and Properties in User Defined Outputs

Computation of Traces Based UDO Calculations

Dimensions Reduction by UDO Calculations

Dynamic Probes

Related Topics

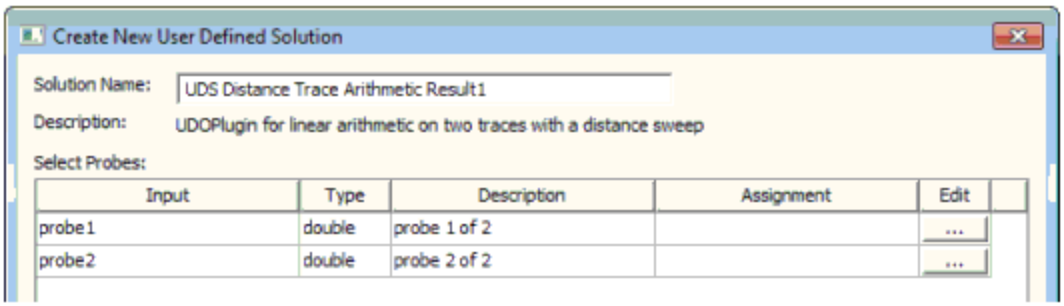
User Defined Outputs: Python Script API

User Defined Outputs: Script Organization

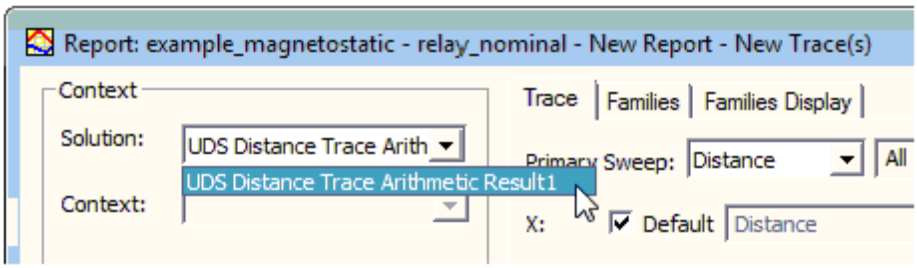
Named Probes and Properties in User Defined Outputs

UDOs allow processing data across traces, solutions and report types. A UDO specifies the named probes and properties for which user selects/enters the values at the time of creation of user defined solution. Probes are very similar to traces except that the user selects the values of only intrinsic variables for probes. The values of design/project variables are selected when a trace is created based upon the user defined solution in reporter.

For example, you could create a user defined solution called UDS Distance Trace Arithmetic Result1.



You can then access this solution in the Reporter.



Related Topics

User Defined Outputs: An Introduction

Computation of Traces Based UDO Calculations

Dimensions Reduction by UDO Calculations

Dynamic Probes

Computation of Traces Based UDO Calculations

When traces that are based upon UDO outputs are computed, the data for probes is computed and passed to the UDO script for each design variation. Along with the probe data, the values of properties entered by user are also passed. The information about the UDO calculations that need to be computed is also made available. The UDO then performs the computation and passes the results to reporter. Note that UDOs can compute and pass back more calculations than have been requested at that point of time. This allows UDOs to compute a set of calculations that take almost same amount of computational resources as any one calculation in that set and cache that with reporter.

Note	When those calculations are subsequently plotted by the user, reporter will use the cached results instead of invoking the computation on UDO.

Related Topics

User Defined Outputs: An Introduction
Named Probes and Properties in User Defined Outputs
Dimensions Reduction by UDO Calculations
Dynamic Probes

Dimensions Reduction by UDO Calculations

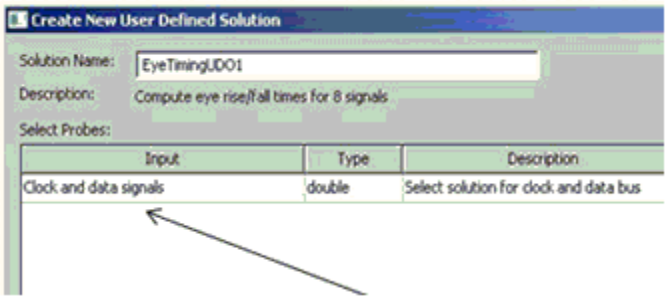
The probes in a UDO can have heterogeneous dimensions of data, for example, one probe in a UDO can have data that is function of **n** intrinsic variable, while another probe in same UDO can have data that is a function of **m** intrinsic variables, with **n** and **m** potentially being different. UDOs allow reducing any number of these intrinsic variables e.g in above example UDO calculations can be function of any number of intrinsic variables including not being function of any intrinsic variable at all. UDO calculations can also be a function an intrinsic variable that none of the probes is function of. **The only restriction is that Freq cannot be reduced if any of the probes are on a Fields report type.**

Related Topics

User Defined Outputs: An Introduction
Named Probes and Properties in User Defined Outputs
Dimensions Reduction by UDO Calculations
Dynamic Probes

Dynamic Probes

In addition to named probes and properties, UDOs can specify named dynamic probes. The difference between probes and dynamic probes is that while the end-user of UDO specifies the complete trace definition for probe, the expression for dynamic probe is specified by UDO code itself and not by the end-user. This allows UDOs to access the data for probes without having the end-user enter each individual probe. For example a UDO can access data for a huge S matrix for a 100 port design without requiring the end-user to enter the probe information for each of those 10,000 quantities. Each dynamic probe is associated with a named probe that is entered by the user; and information about solution, context, and intrinsic variables is used from user-selected probe. However multiple dynamic probes can be associated with the same user-selected probe. The dynamic probes are enquired from UDOs at the time of trace computation and not at the time of creation of user-defined solution.



This means that you select solution, context, and values of intrinsic variables just once; and the same information is used (in this case) for all clock and data signals. The expression for those signals comes from the UDO code.

Related Topics

User Defined Outputs: An Introduction

Named Probes and Properties in User Defined Outputs

Computation of Traces Based UDO Calculations

Dimensions Reduction by UDO Calculations

Dynamic Probes

User Defined Outputs: Python Script API

A User Defined Output (UDO) extension is implemented as an IronPython script that defines a class with a specific name: **UDOExtension** which derives from a specific base class **IUDOPluginExtension** and implements its abstract methods.

UDO Extension IMPLEMENTATION

Optional Functions in IDO Extension Abstract Class

Data Types Used in Python Script

Working With Properties for UDO

Other Application Specific Classes Used in Python Scripts

User Defined Outputs: Messaging Methods

Using .NET Collection Classes and Interfaces in Python Scripts

Related Topics

User Defined Outputs: An Introduction

User Defined Outputs: Script Organization

UDO Extension IMPLEMENTATION

The purpose, argument list and expected return types for each of the **IUDOPluginExtension** abstract methods, which the UDO author is expected to implement are described below.

Import Statements

UDOExtension Class

IUDOPluginExtension Abstract Class

Related Topics

User Defined Outputs: Python Script API

Import Statements

The base class to be used and the types it uses in turn are contained in .NET assemblies. The use of these requires that the assemblies be imported into the UDO script: the following import statements should be added to the top of the python script:

```
from Ansys.Ansoft.ModulePluginDotNet.Common.API import *
from Ansys.Ansoft.ModulePluginDotNet.Common.API.Interfaces import *
from Ansys.Ansoft.ModulePluginDotNet.UDO.API.Interfaces import *
from Ansys.Ansoft.ModulePluginDotNet.UDO.API.Data import *
```

Related Topics

UDO Extension IMPLEMENTATION

UDOExtension Class

The UDO itself should be implemented as an IronPython class called ***UDOExtension*** which **must** derive from the ***IUDOPluginExtension*** abstract base class **(**from the **Ansys.Ansoft.ModulePluginDotNet.UDO.API.Interfaces** namespace**)**.

Note that power users could derive a class hierarchy tuned toward a specific type of UDOs and that they can derive from their own base classes. The only requirement is that directly or indirectly, the UDO class must derive from ***IUDOPluginExtension.***

Example:

```
def BaseClassUDO ((IUDOPluginExtension):
#base class implementation
    …
def UDOExtension ((BaseClassUDO):
#UDO class implementation
…
```

Related Topics

UDO Extension IMPLEMENTATION

IUDOPluginExtension Abstract Class

The implementation of the IUDOPluginExtension class will be described in this section using a simple UDO example that expects a single probe and reduces its dimension returning as its outputs, the max, min and average of its input probe data. The script in its entirety will also be listed later on.

Required functions:

The IUDOPluginExtension abstract class declares the following abstract methods that must be implemented in the UDOExtension class or one of its base classes. Not implementing any of these methods will result in a run-time error and a non functioning UDO. The UDS refers to user Defined Solution parameters.

GetUDSName()

GetUDSDescription()

GetUDSSweepNames()

GetCategoryNames()

GetQuantityNames(string categoryName)

GetQuantityInfo(string quantityName)

GetInputUDSParams(List<UDSProbeParams> udsParams,

GetDynamicProbes(List<UDSDynamicProbes> dynamicProbes);

Compute(IUDSInputData inData,

Related Topics

UDO Extension IMPLEMENTATION

GetUDSName()

- **Purpose**: Return a string that is used as a prefix for all solution instances created using this UDO.
- **Returns**: string.

Example:

```
def GetUDSName(self):

return "MinMaxAvg
```

Related Topics

IUDOPluginExtension Abstract Class

GetUDSDescription()

- **Purpose**: Returns a description for the UDO, its purpose etc. This is used in multiple UDO related dialogs in the application to describe the UDO.
- **Returns**: string.

Example:

```
def GetUDSDescription(self):

return "Sample UDO for dimension reducing quantities
```

Related Topics

IUDOPluginExtension Abstract Class

GetUDSSweepNames()

- **Purpose**: Returns a list of sweep names to be used for the solution generated by the UDO. These will appear in the sweeps list displayed in the standard reporter dialog when used to create reports from the solution generated by the UDO.
- **Returns**: list of strings. If the UDO outputs have no sweeps, return the empty list [].

Example:

```
# Returns list of sweeps names
# We have no sweeps as we reduce them.
def GetUDSSweepNames(self):

return []
```

Related Topics

IUDOPluginExtension Abstract Class

GetCategoryNames()

- **Purpose**: The outputs that the UDO solution provides/generates can be classified into multiple categories (like how the application does as displayed in the report creation dialog). These will be listed in the categories box in the dialog when creating reports from the UDO generated solution data.
- **Returns**: list of strings.

Example:

```
def GetCategoryNames(self):

return ["UDOOutputs"]
```

Related Topics

IUDOPluginExtension Abstract Class

GetQuantityNames(string categoryName)

- **Purpose**: For each of the category names returned from the *GetCategoryNames* method, this function is called to return a list of quantities to be organized under that category name. **Note that the quantity names must be unique across the categories: that is, no two categories can have quantities with the same name.**
- **Parameters**:
  - categoryName (input python string) - category name.
- **Returns**: python list of strings.

Example:

```
# returns a list of quantity names for the supplied category name
def GetQuantityNames(self, catName):

if catName == "UDOOutputs":

    return ["min_val", "max_val", "avg_val"]

else:

    return []
```

Related Topics

IUDOPluginExtension Abstract Class

GetQuantityInfo(string quantityName)

- **Purpose**: For each quantity that the UDO creates, it must also describe the quantity (unit and other details). This method is called for each quantity name (across all categories) as returned from an earlier call of the *GetQuantityNames* method.
- **Parameters**:
  - quantityName (input string) - quantity name.
- **Returns**: Object of type QuantityInfo.

Example:

```
# Returns an instance of QuantityInfo for the qtyName supplied or None if such a
# quantity could not be found
def GetQuantityInfo(self, qtyName):
```

```
# All the quantities we have are simple doubles
# we can leave them unitless
return QuantityInfo(Constants.kDoubleParamStr)
```

Related Topics

IUDOPluginExtension Abstract Class

GetInputUDSParams(List<UDSProbeParams> udsParams,

IPropertyList propList,

List<UDSProbeParams> userSelectionForDynamicProbes)

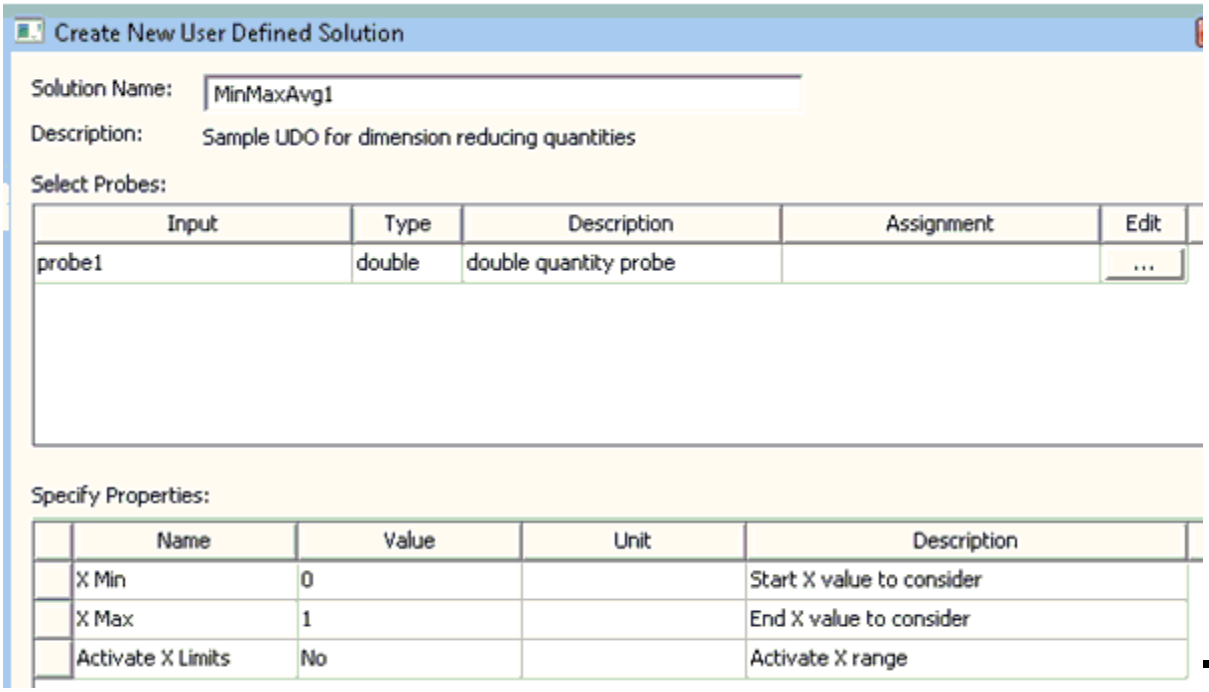
- **Purpose**: This is the main definition part of the UDO. The supplied arguments are used to populate details of the parameters to which the UDO user will specify value, specify the probe names and their types as well as the dynamic probe selections.
- **Parameters**:
  - udsParams - .NET list of UDSProbeParams objects: The UDO script is expected to add one instance of UDSProbeParams for each probe definition it wants displayed. The UDO user will, when creating the UDO solution assign a matching quantity to each such probe.
  - propList - IPropertyList object: The propList object is used to add properties that should be displayed to the user for data collection. These properties with the user supplied values will be returned to the UDO script in the Compute methods.
  - userSelectionForDynamicProbes - .NET list of UDSProbeParam objects.
- **Returns**: boolean: True on success, False on failure.

Example:

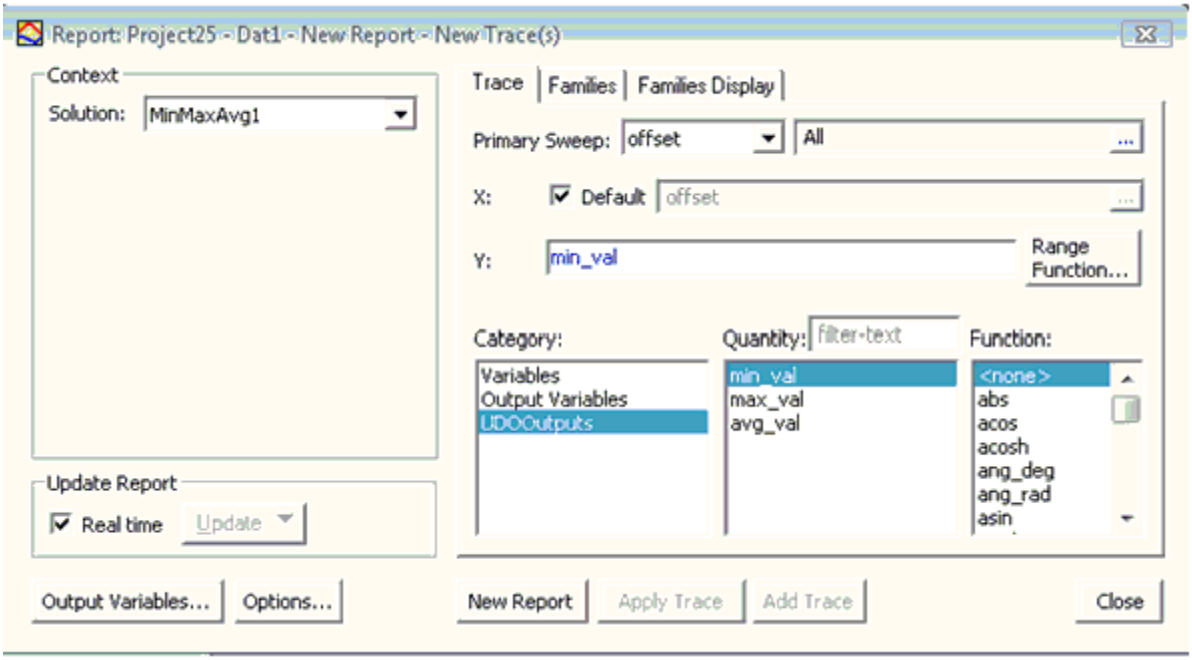
```
# Returns list of UDSParams and list of dynamic properties
# Adds setup time properties to the propList
def GetInputUDSParams(self, udsParams, propList, userSelectedDynamicProbes):


# Add the probes. We need only one double quantity
param1 = UDSProbeParams("probe1",
  "double quantity probe",
  Constants.kDoubleParamStr,
  "", "")
udsParams.Add(param1)


# Add the properties we want the user to supply
```

```
    # In this case, we will ask for a start/end range for
    # X parameters. Since we cannot reasonably provide defaults
    # as we have no idea what the sweep limits will be, we will
    # also ask if the limits are to be activated.
    prop = propList.AddNumberProperty("X Min", "0")
    prop.Description = "Start X value to consider"

    prop = propList.AddNumberProperty("X Max", "1")
    prop.Description = "End X value to consider"

    # For menus, the first option is the default.
    prop = propList.AddMenuProperty("Activate X Limits", ["No", "Yes"])
    prop.Description = "Activate X range"

    return True
```

The above function results in the following dialog when you click **Reports>Create User Defined Solution**. The mapping from the UDSParams and the properties to the GUI elements should be unambiguous. The name and description of the UDS are also displayed in this dialog.



When a report is created from the UDO dialog box, the category/quantity names specified by the UDO are used (as shown below).



Related Topics

IUDOPluginExtension Abstract Class

GetDynamicProbes(List<UDSDynamicProbes> dynamicProbes);

- **Parameters**: dynamicProbes - .Net list of UDSDynamicProbes objects. Output parameter.
- **Returns**: True on success, False on failure.

Example:

```
# Returns list of UDSParams and list of dynamic properties
# output UDSDynamicProbeCollection probes
def GetDynamicProbes(self, probes):
pass
```

Related Topics

IUDOPluginExtension Abstract Class

Compute(IUDSInputData inData,

IUDSOutputData outData,

IPropertyList propList,

IProgressMonitor progressMonitor)

- **Purpose**: This is the main computation method which generates the data for the quantities that make up the UDO solution.
- **Parameters**:
  - ***inData*** – UDSInputData object: Used to get the input probe data.
  - ***outData*** – UDSOutputData object: Used to set the UDO solution quantity and sweep data.
  - ***propList*** – IPropertyList object: Used to get the user entered values for each of the properties defined during the *GetInputUDSParams* call.
  - ***progressMonitor*** – IProgressMonitor object. This can be used to set progress for long running calculations, check for user initiated abort etc.
- **Returns**: True on success, False on failure.

The data is received from UI using ***IUDSInputData*** API***.*** It is processed and the result data is sent to UI using ***IUDSOutputData*** API.

Example:

```
# IUserDefinedSolutionHandle API implementation.
# Calculates output values and sets them using IUDSInputData/IUDSOutputData API.
def Compute(self, inData, outData, propList, progMon):

    # Get the sweeps associated with the probe and validate
    # use the probe name that we had defined earlier
    sweeps = inData.GetSweepNamesForProbe("probe1")
    if( sweeps == None or sweeps.Count > 1):
    AddErrorMessage(self.GetName() + "Unexpected sweep count 0 or > 1 in Compute")
    return False



    # Get the data associated with our probe
    probeData = inData.GetDoubleProbeData("probe1")
    sweepData = inData.GetSweepsDataForProbe("probe1", sweeps[0])

    # Get the user specified properties.
    # Note that ideally, these "X Min" etc names should be written as
```

```
# constant membets and referred to in both the GetInputUDSParams
# and in Compute to reduce the change of typos.
useXRangeProp = propList.GetMenuProperty("Activate X Limits").SelectedMenuChoice
xRangeStart = propList.GetNumberProperty("X Min").ValueSI
xRangeEnd = propList.GetNumberProperty("X Max").ValueSI

# At this stage, one can look at the RequestedQuantities and create
# a dictionary to later check against. However, I am simply computing
# all the quantities.
minVal = 0
maxVal = 0
avgVal = 0

# Check if we need to perform range computation
if useXRangeProp == "Yes":
    seenAny = False
    avgSum = 0
    count = 0


# zip is used since we also need to pull in sweep data
# an index and the array notation could also have been used
for probeVal, sweepVal in zip(probeData, sweepData):
  if sweepVal < xRangeStart or sweepVal > xRangeEnd:
  pass

  # Note that in a better written script, this code would be
  # refactored into it's own function to avoid code
  # duplication
  if not seenAny:
  minVal = probeVal
  maxVal = probeVal
  avgSum = probeVal
  seenAny = True
  count = 1
  else:
  if probeVal < minVal:
```

```
minVal = probeVal

        if probeVal > maxVal:
maxVal = probeVal

        avgSum += probeVal
        count += 1

          if seenAny:
        avgVal = avgSum/count


      else:
          seenAny = False
          avgSum = 0
          for probeVal in probeData:
        if not seenAny:
        minVal = probeVal
        maxVal = probeVal
        avgSum = probeVal
        seenAny = True
        else:
        if probeVal < minVal:
minVal = probeVal

        if probeVal > maxVal:
maxVal = probeVal

        avgSum += probeVal

          if seenAny:
        avgVal = avgSum/probeData.Count



      # Finally set the output values. Note that these are always set as

      # lists even if we have just one item.
```

```
outData.SetDoubleQuantityData("min_val", [minVal])
outData.SetDoubleQuantityData("max_val", [maxVal])
outData.SetDoubleQuantityData("avg_val", [avgVal])

# And we are done.
return True
```

Related Topics

IUDOPluginExtension Abstract Class

Optional Functions in IDO Extension Abstract Class

The following functions, while a part of the IUDOExtension abstract class, have meaningful default implementations and are therefore optional. However, they can be overridden to take advantage of advanced functionality.

Validate(List<string> errorStringList,

Related Topics

User Defined Outputs: Python Script API

Validate(List<string> errorStringList,

List<UDSProbeParams> udsProbParams,

IPropertyList propList,

List<UDSProbeParams> userSelectionForDynamicProbes)

- **Purpose**: This method is used to validate the user choices. The values of the properties entered, the probes etc. can be checked for suitability.
- **Parameters**:
  - udsProbParams - C# list of UDSProbeParams objects.
  - ***propList*** – _IPropertyList object.
  - userSelectionForDynamicProbes - C# list of UDSProbeParams objects.
  - errorStringList - C# list of python strings. Output parameter. Should be set only if validation failed; ignored if validation is successful. One error string should be set per each validation error.
- **Returns**: True on validation success, False on failure.
- **Default implementation**: always returns true.

Example:

```
def Validate(self, errorStringList,probeList,propList, dynamicProbes):
```

```
if probeList == None or probeList.Count == 0:
    errorStringList.Add("Empty probe list")
    return False
return True
```

Related Topics

Optional Functions in IDO Extension Abstract Class

Data Types Used in Python Script

There are several types that you must use while authoring the python script. Some of them are used to pass data from UI to python script and to provide interface for working with this data. Some are used to pass data from python script to UI.

To pass data from python script to UI the objects of the C# class must be created in python script using their C# constructors. Then they can be set as functions return values or set to the output parameters using their API.

Constants class

kTraceTypeStr : string constant used to specify an input of trace type

kSolutionTypeStr : string constant used to specify an input of solution type

kNumberTypeStr : string constant used to specify an input of number type

kTextTypeStr : string constant used to specify an input of text type

kBoolTypeStr : string constant used to specify an input of boolean type

kStandardReportStr : string constant to specify a standard report

kEyeDiagramReportStr : string constant to specify an eye diagram report

kUserDefinedReportStr : string constant to specify a user defined report

kSweepDomainStr : string constant to specify the sweep domain

kTimeDomainStr : string constant to specify the time domain

UDDInputParams class

The objects of this class must be created in python script in the **GetUDDInputParams()** function and the **SetUDDInputParams()** function.

Attributes :

Input Name (string)

Input Description (string)

Input Type (Can be Boolean, Number, Text, Trace or Solution) (string)

BoolData (boolean)

DoubleData (double)

TextData (string)

ReportType (string)

SolutionName (string)

DomainName (string)

Constructors:

UDDInputParams(string name, string description, string type)

UDDInputParams(string name, string description, string type, bool data)

UDDInputParams(string name, string description, string type, double data)

UDDInputParams(string name, string description, string type, string data)

UDDInputParams(string name, string description, string type, string reportType, string solutionName, string domainName)

Property Accessors :

Name : Get/Set the name of an input

Description : Get/Set the desccription of an input

Type : Get/Set the type of an input

BoolData : Get/Set the data of a boolean input

DoubleData : Get/Set the data of a number input

TextData : Get/Set the data of a text input

ReportType : Get/Set the report type

SolutionName : Get/Set the name of the solution

DomainName : Get/Set the name of the domain

IProgressMonitor Abstract Class

The object of this class is a progress monitor. It is used to display calculations progress in UI and check is the user has requested an abort of the computation.

When displayed in the application, each progress message has four items:

A task name

A sub-task name

The progress amount

A button to abort the task in progress.

All of this functionality and abort interaction is achieved using the following functions.

SetTaskName (string taskName):

SetSubTaskName (string subTaskName)

BeginTask (string name)

SetTaskProgressPercentage(int progressPercent)

CheckForAbort(): If the quantities being generated are computationally expensive, the UDO author can periodically call this method and then call EndTask with Fail and return False.

EndTask (bool passFail)

Example:

```
progMon.BeginTask("Process DQS")
progMon.SetSubTaskName("Compute UI segments")
progMon.SetTaskProgressPercentage(33)
progMon.SetSubTaskName("Compute the rest")
progMon.SetTaskProgressPercentage(100)
progMon.EndTask(True)
```

IUDSInputData

The purpose of this class is to get data (probe and sweep) from Desktop.

Examples in this section are just to show proper syntax of the function calls. For actual usage of the class see Compute function example.

GetDoubleProbeData(probeName)

GetSweepsDataForProbe(probeName, sweepName)

GetComplexProbeData(probeName)

GetSweepNamesForProbe(probeName)

GetRequiredQuantities()

GetVariableValues()

GetInterpolationOrdersData(probeName);

Related Topics

Data Types Used in Python Script

GetDoubleProbeData(probeName)

- **Purpose**: This is the primary mechanism by which the UDO script obtains the probe data (as double precision values) for its compute process.
- **Parameters**:
  - probeName: string representing the probe name for which data is requested. This has to be one of the many probes supplied during a call to the UDO's **GetInputUDSParams** method.
- **Returns**: .NET double Array of data for the specified probe if the probe exists or null if the probe is unknown.

Example:

```
# doubleData is a list of floats
doubleData = inData.GetDoubleProbeData("probe1")
```

Related Topics

Data Types Used in Python Script

GetSweepsDataForProbe(probeName, sweepName)

- **Purpose**: All probe data that is supplied is associated with one ore more sweep (an intrinsic quantity like Time, Frequency, Theta, Phi etc that is swept) quantities.
- **Parameters**:
  - probeName - probe name for which which want the sweep data
  - sweepName - sweep name
- **Returns**: .NET double Array of data for the specified probe and sweep.

Example:

```
# sweepData is C# Array of doubles (floats in python)
sweepData = inData.GetSweepsDataForProbe("FarFieldsProbe","Freq"])
```

Related Topics

Data Types Used in Python Script

GetComplexProbeData(probeName)

- **Purpose**: The primary mechanism by which the UDO retrieves data for its input probes (if it expects complex data for the probe).
- **Parameters**:
  - probeName - probe name for which complex data is requested
- **Returns**: .NET double Array (float in python) of data for the specified probe. Each pair of floats represent one complex number: first value is for real part, second value for imaginary part. For instance, array [10.0, 0, 5.1, 2.1] represents 2 complex numbers: (10.0, 0) and (5.1, 2.1).

Example:

```
# complexDataAsDouble is C# Array of doubles (floats in python)
# each pair of floats represents one complex number
complexDataAsDouble = inData.GetComplexProbeData("FarFieldsProbe")
# creating a list of complex numbers from complexDataAsDouble array
complexData = []
if complexDataAsDouble != None:
    for i in xrange(0,complexDataAsDouble.Count , 2):
        complexData.append(complex(complexDataAsDouble[i],complexDataAsDouble[i+1]))
```

Related Topics

Data Types Used in Python Script

GetSweepNamesForProbe(probeName)

- **Purpose**: To obtain the list of sweep quantity names associated with a given probe. This also indicates the dimensionality of the data. One name implies that the probe-data is 2D (probe-quantity vs Sweep Quantity) and two names implies 3D data (probe-quantity vs Sweep 1 X Sweep 2).
- **Parameters**:
  - probeName - probe name.
- **Returns**: .NET IList<string> - list of sweep names for the current probe name.

Example:

```
# sweepNames is C# Array of strings
sweepNames = inData.GetSweepNamesForProbe("FarFieldsProbe")
```

Related Topics

Data Types Used in Python Script

GetRequiredQuantities()

- **Purpose**: A given UDO can specify that it provides one of more computed quantities. The user might choose to create a report from only a few among the various available UDO outputs. This function, returns that list of the UDO output quantities that the user has requested. Only these need be computed in the UDO's compute method.
- **Returns**: .NET IList<string> - list of required quantities names.

Example:

```
# quantities is C# Array of strings
quantities= inData.GetRequiredQuantities()
```

Related Topics

Data Types Used in Python Script

GetVariableValues()

- **Purpose**: This allows the UDO to obtain the names and values of all the design variables for which the UDO quantities are being requested.
- **Returns**: .NET IDictionary<string,string> of key-value pairs for variables. Both key and value are strings.

Example:

```
# theDict is C# Dictionary<string, string>
theDict = inData.GetVariableValues()
if theDict != None:
#varPair is of .Net KeyValuePair type
for varPair in theDict:
    varName = varPair.Key #string
    varValue = varPair.Value #string
```

Related Topics

Data Types Used in Python Script

GetInterpolationOrdersData(probeName);

- **Purpose**: Returns the interpolation orders that are associated with the probe-data. The probe data is specified at each value of the various sweeps. Any value in between the sweep data points, can use the interpolation data to get a possibly more accurate (compared to linear interpolation) inter-sweep value.
- **Parameters**:
  - probeName (input python string) - probe name.
- **Returns**: NET byte Array of interpolation order for the specified probe. These are to be treated as 8bit signed integers, that is, their values range from 0-127.

Example:

```
# interData is C# Array of bytes (integers in python)
interData = inData.GetInterpolationOrdersData(kProbeNames[0])
for interValue in theDict:
    order = interValue # interValue and order are integers
```

Related Topics

Data Types Used in Python Script

IUDSOutputData

This type is a twin of the IUDSInputData in that it is used to store the values computed by the UDO's compute method.

Examples in this section are just to show proper syntaxis function calls. For actual usage of the class see the Compute function example.

SetSweepsData(sweepName, sweepData)

SetDoubleQuantityData(qtyName,qtyData)

SetComplexQuantityData(qtyName, qtyData)

Related Topics

Data Types Used in Python Script

SetSweepsData(sweepName, sweepData)

- **Purpose**: Each quantity that is computed by the UDO can be associated with a sweep. If it is, the values that make up the sweep's data points must be specified using this call.
- **Parameters**:
  - sweepName (string) - sweep name.
  - sweepData (python list of floats) - sweep data for the specified sweep.
- **Returns**: True on success, False on failure.

Example:

```
sweepList = [12.3, 14.5, 16.7]
outData.SetSweepsData("Freq", sweepList)
```

Related Topics

Data Types Used in Python Script

SetDoubleQuantityData(qtyName,qtyData)

- **Purpose**: This method is used to record the computed quantity data for each output that is computed. Please note that unless all the sweeps are reduced, this should be used in conjunction with SetSweepsData
- **Parameters**:
  - qtyName (string) - quantity name.
  - qtyData (python list of floats) - quantity data for the specified quantity.
- **Returns**: True on success, False on failure.

Example:

```
doubleList = [12.3, 14.5, 16.7]
outData.SetDoubleQuantityData("V1PlusV2", doubleList)
```

Related Topics

Data Types Used in Python Script

SetComplexQuantityData(qtyName, qtyData)

- **Purpose**: If the quantity computed is a complex quantity, use this method to set the quantity values. Any sweep values must be set separately via the SetSweepsData method.
- **Parameters**:
  - qtyName (string) - quantity name.
  - qtyData (python list of floats) - quantity data for the specified quantity. Complex numbers are passed as pairs of floats
- **Returns**: True on success, False on failure.

Example:

```
doubleFromComplexList=[]
complexList = [(1+1j), (2+4j), (9.1+3.2j)]
for aComplex in complexList:
doubleFromComplexList.append(aComplex.imag)
doubleFromComplexList.append(aComplex.real)
outData. SetComplexQuantityData ("V1PlusV2", doubleFromComplexList)
```

Related Topics

Data Types Used in Python Script

Working With Properties for UDO

A property is the unit for collecting and using input from the user that is used to influence the UDO's Compute. These are initially set up when the UDOs **GetInputUDSParams** method is called and are retrieved in the UDO's Compute method.

There are 3 supported property types that could be used in the UDO script:

- INumberProperty to specify number properties (with unit support).
- IMenuProperty to allow the user to select from a list of options.
- ITextProperty to allow the user to enter text.

The IPropertyList type implements a collection for these properties.

IPropertyList Abstract class

IProperty Abstract class

INumberProperty Abstract class

ITextProperty Abstract class

IMenuProperty Abstract class

Related Topics

User Defined Outputs: Python Script API

IPropertyList Abstract class

Attributes:

- AllProperties (IEnumerable<IProperty> - see *IProperty*)
- NumProperties (int)

Functions:

- GetProperty(string propName): Returns a named property as an IProperty.
- GetMenuProperty (string propName): Returns the named property as an IMenuProperty.
- GetTextProperty (string propName): Returns the named property as an ITextProperty
- GetNumberProperty (string propName): Returns the named property as an INumberProperty
- DeleteProperty (string propName): Deletes an already added named property
- AddNumberProperty(string name, string numberWithUnits): Adds a new number property. If a property with the same name already exists, it is overwritten.
- AddTextProperty(string name, string textValue): Adds a new named text property with the supplied value. Any existing property with the same name is overwritten.
- AddMenuProperty(string name, IList<string> menuChoices): Creates a new named menu property with the supplied list of choices. The default selection is set to item 0 (the first item). Any property with the same name is overwritten.

IProperty Abstract class

Attributes:

- Name (string)
- Description (string)
- PropType (read-only EPropType - see Constants)

Constructor:

- IProperty(string name, EPropType type)

The class is used as base class for INumberProperty, IMenuProperty, and ITextProperty.

INumberProperty Abstract class

Base class:

- abstract class IProperty

Attributes:

- ValueSI (read-only double)
- ValueInUnits (read-only double)
- Units (read-only string)
- HasUnits (read-only bool)

Constructor:

- INumberProperty(string name)

Functions:

- Set(string numberWithUnits)
- SetDouble(double number, string unitString)

ITextProperty Abstract class

Base class:

- abstract class IProperty

Attributes:

- Text (string)

Constructor:

- ITextProperty(string name)

IMenuProperty Abstract class

Base class:

- abstract class IProperty

Attributes:

- MenuSelection (int): *This represents the index into the MenuChoices list.*
- SelectedMenuChoice (string): *This is the item in the MenuChoices list corresponding to the MenuSelection index*
- MenuChoices (IList<string>)

Constructor:

- IMenuProperty (string name)

Example:

```
# adding data to IPropertyList propList; used in Compute function
prop = propList.AddNumberProperty('Offset 1', '0')
prop.Description = 'Trace 1 Offset'
prop = propList.AddNumberProperty("TRATE", "800 MHz")
prop.Description = "Frequency"
prop = propList.AddTextProperty("Text", "The Text")
prop.Description = "Text Property"
prop = propList.AddMenuProperty('Operation', ['Add', 'Subtract', 'Max' ,'Min','Mean'])
prop.Description = 'Operation menu'

# reading data from IPropertyList propList; used in Validate function
numOfNumberProperties = 0
if propList != None and propList.AllProperties != None:
```

```
    for prop in propList.AllProperties:
if prop.PropType == Constants.EPropType.PT_NUMBER:
numOfNumberProperties ++
```

Other Application Specific Classes Used in Python Scripts

This section describes other classes used in Python scripts:

Constants Class

UDSProbeParams Class

UDSDynamicProbes Class

QuantityInfo Class

IProgressMonitor Abstract Class

Related Topics

User Defined Outputs: Python Script API

Constants Class

The constants used in python script are defined in the Constants class.

Attributes:

- kDoubleParamStr : string constant used to specify *double* as the type of a quantity
- kComplexParamStr: string constant used to specify *complex* as the type of a quantity

Enum EPropType: (used to set property type)

```
EPropType.PT_NUMBER
 EPropType.PT_TEXT
 EPropType.PT_MENU
```

Example:

```
paramType = Constants.kDoubleParamStr
propType = Constants.EPropType.PT_NUMBER
```

Related Topics

Other Application Specific Classes Used in Python Scripts

UDSProbeParams Class

The objects of this class must be created in python script in **GetInputUDSParams** function**.** They are supplied to the Validate function if implemented.

Attributes:

- ProbeName (read-only string)
- ProbeDescription (read-only string)
- ParamType (read-only string)
- ReportTypeName (read-only string)
- ComponentExpression (read-only string)

Constructor: UDSProbeParams(string probeName, string probeDescription, string paramType, string reportTypeName, string componentExpression);

- probeName - required.
- probeDescription - optional (can be empty string).
- paramType - required; can be one of the Constants
  - kDoubleParamStr
  - kComplexParamStr
- reportTypeName - optional (can be empty string)
- ComponentExpression - optional (can be empty string)

Example:

```
udsProbParam = UDSProbeParams("probe1","", Constants.kDoubleParamStr, "", "",)
```

Related Topics

Other Application Specific Classes Used in Python Scripts

UDSDynamicProbes Class

Attributes:

- UDSParam (read-only UDSProbeParams)
- UserSelectedProbeName (read-only string)

Constructor: UDSDynamicProbes (UDSProbeParams udsParam, string userSelectedProbeName.

- udsParam - required
- userSelectedProbeName - required

Example:

```
udsProbParam = UDSProbeParams("probe1","", Constants.kDoubleParamStr, "", "",)
selectedName = "probe1"
udsDynamicProbParam = UDSDynamicProbes(udsProbParam , selectedName )
```

Related Topics

Other Application Specific Classes Used in Python Scripts

QuantityInfo Class

Attributes:

- ParamType (read-only string)
- FullUnitType (read-only string)

Constructors:

- QuantityInfo(string paramType)
- QuantityInfo(string paramType, string fullUnitType)
- **Parameters**:
  - paramType can be one of the Constants

    kDoubleParamStr

    kComplexParamStr

- fullUnitType is a case insensitive string representing full unit type. It is not defined in Constants. Instead you can use any of the units in string representation - for example, "mm" or"ghz".

Example:

```
quantityInfo1 = QuantityInfo(Constants.kDoubleParamStr)
quantityInfo2 = QuantityInfo(Constants.kDoubleParamStr,"ghz")
```

Related Topics

Other Application Specific Classes Used in Python Scripts

IProgressMonitor Abstract Class

The object of this class is a progress monitor. It is used to display calculations progress in UI and check is the user has requested an abort of the computation.

When displayed in the application, each progress message has four items:

- A task name
- A sub-task name
- The progress amount
- A button to abort the task in progress.

All of this functionality and abort interaction is achieved using the following functions.

- SetTaskName (string taskName):
- SetSubTaskName (string subTaskName)
- BeginTask (string name)
- SetTaskProgressPercentage(int progressPercent)
- CheckForAbort(): If the quantities being generated are computationally expensive, the UDO author can periodically call this method and then call EndTask with Fail and return False.
- EndTask (bool passFail)

Example:

```
progMon.BeginTask("Process DQS")
progMon.SetSubTaskName("Compute UI segments")
progMon.SetTaskProgressPercentage(33)
progMon.SetSubTaskName("Compute the rest")
progMon.SetTaskProgressPercentage(100)
progMon.EndTask(True)
```

Related Topics

Other Application Specific Classes Used in Python Scripts

Using .NET Collection Classes and Interfaces in Python Scripts

Some of the API functions specified above use .Net collection classes and interfaces, that is, Array class, IList interface, IEnumerable interface, and IDictionary interface. The following section describes how to work with the .Net collection objects in Python scripts.

.NET Array, IEnumerable, and IList objects can be indexed and iterated over as if they were Python lists. You can also check for membership using 'in'. To get .Net Array and IList sizes you can use python's 'len' or .Net 'Count'.

Example:

Getting size:

```
arraySize = doubleDataArray.Count
arraySize = len(doubleDataArray)


listSize = sweepsNamesList.Count
listSize = len(sweepsNamesList)
```

Iterating:

```
for sweep in sweepsNamesList:
    print sweep

for in in xrange(listSize)
    print sweepsNamesList[i]
```

Checking for membership:

```
if 'Time' in sweepsNamesList:
    doThis()
```

```
else:
    doThat()
```

For .NET IDictionary, the same as for Array and IList, you can get size with 'len' or 'Count' and check for membership of the keys using 'in'. Getting values for the keys also works the same way as in python 'dict'.

Example

Getting size:

```
varValuesSize = varValues.Count
varValuesSize = len(varValues)
```

Checking for membership:

```
if 'offset' in varValues:
    print varValues['offset']
```

Getting value:

```
if 'offset' in varValues:
    offsetValue = varValues['offset']
```

As for iteration .NET Dictionary is different from python dict. While iterating, python dict will return keys, .Net Dictionary will return .Net KeyValuePair.

Example:

Iterating:

for .Net IDictionary:

```
for varPair in varValues: #varPair is of .Net KeyValuePair type
    varName = varPair.Key
    varValue = varPair.Value
```

for python dict:

```
for varName in varValues:
    varValue = varValues[varName]
```

You can use python types instead of .Net types if you prefer. For this you need to cast .Net Array and .Net iList to python list type and .Net Dictionary to python dict type.

Casting should not be used for data arrays - it can be extremely costly for the memory usage as well as time consuming.

Example:

```
aPythonList = list(dotNetArray)
```

```
    aPythonList = list(dotNetList)
    aPythonDict = dict(dotNetDictionary)
```

Related Topics

User Defined Outputs: Python Script API

User Defined Outputs: Messaging Methods

Messaging methods are provided to convey additional information to the user from any of the UDOs methods. The Compute function is the one typically location where such use is anticipated. Any message sent via these functions are displayed in the application's message window using the appropriate icon.

These functions can also be used for debugging purposes.

- **AddErrorMessage(string)**: Call this method to convey an error condition to the user.
- **AddWarningMessage(string)**: Call this method to convey a warning message: typically used for conditions that are not ideal but can be tolerated by the script.
- **AddInfoMessage(string)**: Call this method to convey an informational message to the user. This is the call to use when outputting messages for debugging purposes.

```
############################################################
# Imports
############################################################
from Ansys.Ansoft.ModulePluginDotNet.Common.API import *
from Ansys.Ansoft.ModulePluginDotNet.Common.API.Interfaces import *
from Ansys.Ansoft.ModulePluginDotNet.UDO.API.Interfaces import *
from Ansys.Ansoft.ModulePluginDotNet.UDO.API.Data import *
class UDOExtension(IUDOPluginExtension):
    def __init__(self):
    pass


    #--- IDA IUDOPluginExtension ----------------------
    def GetUDSName(self):
    return "MinMaxAvg"


    #--- ISA IUDOPluginExtension ----------------------
    def GetUDSDescription(self):
    return "Sample UDO for dimension reducing quantities"
```

```
#--- ISA IUDOPluginExtension ----------
# Returns list of category names
def GetCategoryNames(self):

return ["UDOOutputs"]

#--- ISA IUDOPluginExtension ----------
# returns a list of quantity names for the supplied category name
def GetQuantityNames(self, catName):

if catName == "UDOOutputs":

    return ["min_val", "max_val", "avg_val"]

else:

    return []


#--- ISA IUDOPluginExtension ----------
# Returns an instance of QuantityInfo for the qtyName supplied or None if such a
# quantity could not be found
def GetQuantityInfo(self, qtyName):

# All the quantities we have are simple doubles

# we can leave them unitless

return QuantityInfo(Constants.kDoubleParamStr)


#--- ISA IUDOPluginExtension ----------
# Returns list of UDSParams and list of dynamic properties
# Adds setup time properties to the propList
def GetInputUDSParams(self, udsParams, propList, userSelectedDynamicProbes):


# Add the probes. We need only one double quantity

param1 = UDSProbeParams("probe1",

    "double quantity probe",
    Constants.kDoubleParamStr,
    "", "")

udsParams.Add(param1)
```

```
# Add the properties we want the user to supply
# In this case, we will ask for a start/end range for
# X parameters. Since we cannot reasonably provide defaults
# as we have no idea what the sweep limits will be, we will
# also ask if the limits are to be activated.
prop = propList.AddNumberProperty("X Min", "0")
prop.Description = "Start X value to consider"

prop = propList.AddNumberProperty("X Max", "1")
prop.Description = "End X value to consider"

# For menus, the first option is the default.
prop = propList.AddMenuProperty("Activate X Limits", ["No", "Yes"])
prop.Description = "Activate X range"

return True

#--- ISA IUDOPluginExtension ----------------------
# Returns list of UDSParams and list of dynamic properties
# output UDSDynamicProbeCollection probes
def GetDynamicProbes(self, probes):
pass

#--- ISA IUDOPluginExtension ----------------------
# Returns list of sweeps names
# We have no sweeps as we reduce them.
def GetUDSSweepNames(self):
return []
```

```
#---------------------------------------------------------------------------
# IUserDefinedSolutionHandle API implementation.
# Calculates output values and sets them using IUDSInputData/IUDSOutputData API.
def Compute(self, inData, outData, propList, progMon):


# Get the sweeps associated with the probe and validate

# use the probe name that we had defined earlier

sweeps = inData.GetSweepNamesForProbe("probe1")

if( sweeps == None or sweeps.Count > 1):

    AddErrorMessage(self.GetName() + "Unexpected sweep count 0 or > 1 in Compute")
    return False



# Get the data associated with our probe

probeData = inData.GetDoubleProbeData("probe1")

sweepData = inData.GetSweepsDataForProbe("probe1", sweeps[0])


    # Get the user specified properties.
    # Note that ideally, these "X Min" etc names should be written as
    # constant members and referred to in both the GetInputUDSParams
    # and in Compute to reduce the change of typos.
    useXRangeProp = propList.GetMenuProperty("Activate X
    Limits").SelectedMenuChoice
    xRangeStart = propList.GetNumberProperty("X Min").ValueSI
    xRangeEnd = propList.GetNumberProperty("X Max").ValueSI

    # At this stage, one can look at the RequestedQuantities and create
    # a dictionary to later check against. However, I am simply computing
    # all the quantities.
  minVal = 0
  maxVal = 0
  avgVal = 0
```

```
        # Check if we need to perform range computation
        if useXRangeProp == "Yes":
seenAny = False
avgSum = 0
count = 0

# zip is used since we also need to pull in sweep data
# an index and the array notation could also have been used
for probeVal, sweepVal in zip(probeData, sweepData):
if sweepVal < xRangeStart or sweepVal > xRangeEnd:
pass

# Note that in a better written script, this code would be
# refactored into its own function to avoid code
# duplication
if not seenAny:
minVal = probeVal
maxVal = probeVal
avgSum = probeVal
seenAny = True
count = 1
else:
if probeVal < minVal:
minVal = probeVal

if probeVal > maxVal:
maxVal = probeVal


avgSum += probeVal
count += 1

if seenAny:
avgVal = avgSum/count

        else:
seenAny = False
```

```
avgSum = 0
for probeVal in probeData:
if not seenAny:
minVal = probeVal
maxVal = probeVal
avgSum = probeVal
seenAny = True
else:
if probeVal < minVal:
minVal = probeVal

if probeVal > maxVal:
maxVal = probeVal

avgSum += probeVal

if seenAny:
avgVal = avgSum/probeData.Count

        # Finally set the output values. Note that these are always set as
        # lists even if we have just one item.
        outData.SetDoubleQuantityData("min_val", [minVal])
        outData.SetDoubleQuantityData("max_val", [maxVal])
        outData.SetDoubleQuantityData("avg_val", [avgVal])

        # And we are done.
        return True
```

Related Topics

User Defined Outputs: Python Script API

User Defined Outputs: Script Organization

As described in the Introduction section, the UDO scripts should all reside under the **UserDefinedOutputs** folder under either of the three library locations (system, user or personal).

Using Script Libraries

Additional .NET Assemblies

Related Topics

User Defined Outputs: An Introduction

User Defined Outputs: Python Script API

Using Script Libraries

If you decide that you need base classes, additional data files, and etc., to organize your UDOs better, you can do so. This type of library organization allows code reuse between similar UDOs and can be very helpful. There is special support provided for this type of script-library organization:

- **All script-library and other support files need to be in a *Lib* sub-directory under the UserDefinedOutputs directory**. Any *.py* files found in such **Lib** directories are ignored and not displayed in the GUI as a valid UDO choice.
- For a UDO script at any given directory depth, all Lib directories in its parent directories will be automatically added to the system include path (and so, any support script files from any Lib directory till the top level UserDefinedOutputs directory can be imported)

Related Topics

User Defined Outputs: Script Organization

Additional .NET Assemblies

Using additional .NET assemblies

Because the UDO functionality uses IronPython, we have access to the full .NET eco system. If needed, any subset of the UDO functionality can be implemented in any .NET language and used by the UDO script. There are simple rules to follow to achieve this.

1. Build your .NET assembly for .NET 2.0 runtime.
2. Drop the built assembly in any **Lib** directory upstream of the UDO script location: that is, if you have your UDO script in *C:\Users\x\PersonalLib\UserDefinedOutputs\a\b\c\myudo.py* and have a .NET assembly called *com.Acme.UDOLib* You can keep the .NET assembly under
   - UserDefinedOutputs\Lib,
   - UserDefinedOutputs \a\Lib,
   - UserDefinedOutputs \a\b\Lib
   - UserDefinedOutputs\a\b\c\Lib
3. Add the following line to your python script
   - Import clr
   - clr.AddReference("com.Acme.UDOLib")
   - import com.Acme.UDOLib -or-- from com.Acme.UDOLib import * *etc*

If for some reason you cannot place the .NET assemblies into a Lib directory under UserDefinedOutputs, you need to do a couple more steps before step 3 listed above.

```
Import sys
sys.path.append("full path to your .NET assembly
location")
```

Related Topics

User Defined Outputs: Script Organization

Using Script Libraries

Toolkit

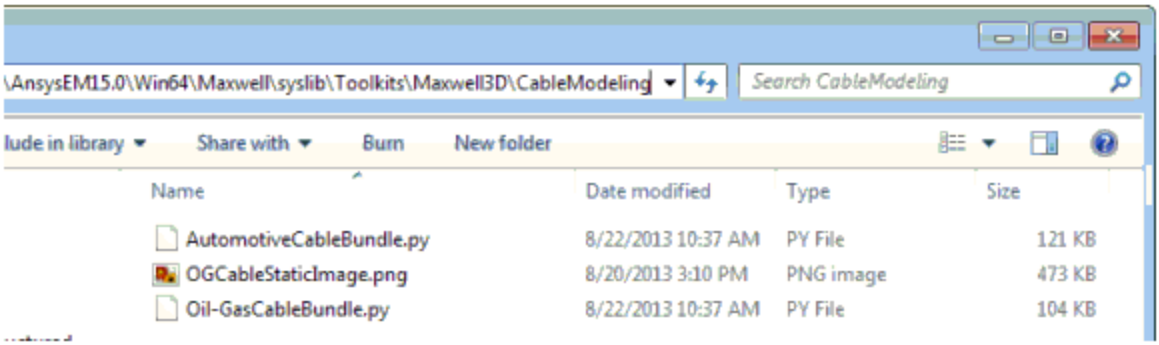
The Toolkit command in the Maxwell menu provides access to a design type-specific IronPython script or a module-specific task. The python script provides a default but customizable UI and canned automations. The UDD format can be adjusted/extended as needed. The default UI is a modal dialog.

This allows Maxwell the flexibility to support changing standards or multiple standards. Similarly, you can create new UDDs to customize the test report.

UDDs are saved in the project result directory. This type of report can be created prior to simulation, but will only be populated when there is a solution.

Undo/Redo/Scripting is supported. If a script includes multiple commands, it takes multiple undo (s) to revert all the changes made by the python script. Also, note that the script playback might not work if the baseline of the design has changed.

A directory called "Toolkits" appears in the syslib, userlib, and personallib of the Maxwell installation. You can add product specific-directories (Maxwell2D, Maxwell3D, RMxprt) in Toolkits, and each directory contains toolkits (scripts) that are specific for that product.



Use the **Maxwell>Toolkit>Update Menu** command to add newly added toolkit scripts to the Toolkit menu. Menu items for files found in <installation>/syslib will be inserted at the first level of the menu.

Related Topics

Toolkit Examples

Toolkit Examples

Maxwell includes the following toolkits for generating 2D and 3D cable bundle models:

- AutomotiveCableBundle
- Oil-GasCableBundle

Automotive Cable Bundle Toolkit

To work with the Automotive Cable Bundle toolkit:

1. Click **Maxwell 2D** (or **Maxwell 3D**)**>Toolkit>CableModeling>AutomotiveCableBundle**.

The **Cable Modeling - Automotive** dialog box appears with a default set of **Cable Parameters** listed.

2. Type the **Name** of the toolkit, and select the **Units** to be used for displayed values.
3. Modify the **Cable Parameters**:
   a. Select the wire **Standard**. You can choose to use either **ISO** or **AWG** standard. The **Wire Type** and **Insulation Type** fields are updated with menus of values corresponding to the chosen standard.
   b. Select the **Wire Type**. The **Conductor Diameter** parameter updates with the corresponding standard value.
   c. Optionally, modify the number of wires (**#Wires**) in the bundle that have this row's set of parameters
   d. Optionally, modify the **Conductor Diameter**.
   e. Select the **Conductor Material**.
   f. Select the **Insulation Type**.
   g. Optionally, modify the **Insulation Thickness**.
   h. Optionally, modify the **Insulation Material**.
4. To add additional cable types, click the **+** button. A row is added in which you can define a new set of cable parameters. Add as many rows as needed to create a bundle consisting of several different types of cable.
5. If you have previously imported a customized set of parameters, you can click **Use Defaults** to use the default values.
6. To export the cross-section parameters into a .csv or .tab file, click **Export**. The **Export Cable Parameters** dialog box appears. Specify the **File name** and **type** (either comma delimited **.csv** or tab delimited **.tab**), and click **Save**.
7. To read in either a comma delimited **.csv** or tab delimited **.tab** file and populate parameters in the toolkit panel, click **Import**. The **Import Cable Parameters** dialog box appears.

Specify the **File name** and **type**, then click **Open**.

8. Specify the **Outer Jacket Parameters**: for the jacket that will encase the cable bundle.
   a. Specify the **Thickness** value for the outer jacket.
   b. Specify the **Material** to be used for the outer jacket.
   c. Specify the **Seeding for wire arrangement** value. The value entered here is the seed value for the pseudo-random number generator used in the conductor packing compute process. Changing this value results in different arrangements of the conductors. This can be useful for performing statistical analysis of the cables.
   d. When finished specifying all of the cable and outer jacket parameters for the bundle, click **Compute** to calculate the minimal bundle diameter of the cable bundle.

Note	The wire radii are adjusted downward very slightly (on the order of 0.1%) at the end of the packing process to eliminate any residual overlaps between the wires and to make the model easier to mesh.

Note	The Circle Packing algorithm is used to compute the values. It tries to automatically pack the conductors in the bundle tightly with a minimum of empty space between them. It generates and tests a large number of random variations in the conductor positions and attempts to minimize the overall diameter of the bundle.

1. To add new variables, click **Variables**. This opens the **Edit Variables** dialog box. All variables that are already present in the project and design levels are listed. Click **Add** to add a new row to create new variables.
2. Optionally, for 3D designs only, click **Extrude**. (The **Extrude** button is not present for 2D designs.) The **Extrude to 3D** dialog box appears.
   - Click **Sweep Along Vector** to extrude along the z axis. Z length is the input parameter.
   - Click **Sweep Along Path** to extrude along a selected path.

Note	3D models of cables are hard to solve if the ratio of extruded length to cross-section diameter is high.

3. Click **Draw** to draw the 2D or 3D cross-section geometry. A validation check is run. This catches errors like any variable missing a value, or any parameter missing a value. For 3D designs, if you chose to **Extrude**, the 3D geometry is drawn.

Oil-Gas Cable Bundle Toolkit

To work with toolkits in the oil and gas industries:

1. Click **Maxwell 2D** (or **Maxwell 3D**)**>Toolkit>CableModeling>Oil-GasCableBundle**.

The **Cable Modeling - Oil & Gas** dialog box appears.

2. Type the **Name** of the toolkit, and select the **Units** to be used for displayed values.
3. Modify the various **Cable Parameters** as needed.
4. If you have previously imported a customized set of parameters, you can click **Use Defaults** to use the default values.

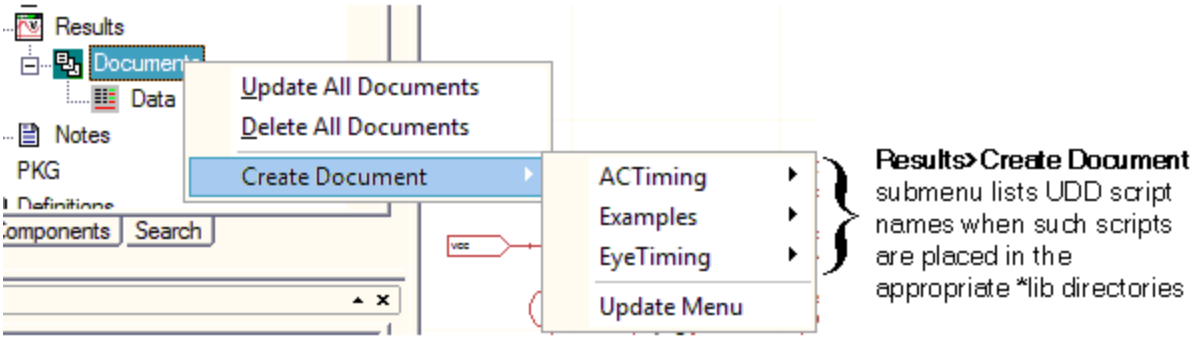
5. To export the cross-section parameters into a .csv or .tab file, click **Export**. The **Export Cable Parameters** dialog box appears. Specify the **File name** and **type** (either comma delimited **.csv** or tab delimited **.tab**), and click **Save**.
6. To read in either a comma delimited **.csv** or tab delimited **.tab** file and populate parameters in the toolkit panel, click **Import**. The **Import Cable Parameters** dialog box appears. Specify the **File name** and **type**, then click **Open**.
7. To add new variables, click **Variables**. This opens the **Edit Variables** dialog box. All variables that are already present in the project and design levels are listed. Click **Add** to add a new row to create new variables.
8. Optionally, for 3D designs only, click **Extrude**. (The **Extrude** button is not present for 2D designs.) The **Extrude to 3D** dialog box appears.
   - Click **Sweep Along Vector** to extrude along the z axis. Z length is the input parameter.
   - Click **Sweep Along Path** to extrude along a selected path.

Note	3D models of cables are hard to solve if the ratio of extruded length to cross-section diameter is high.

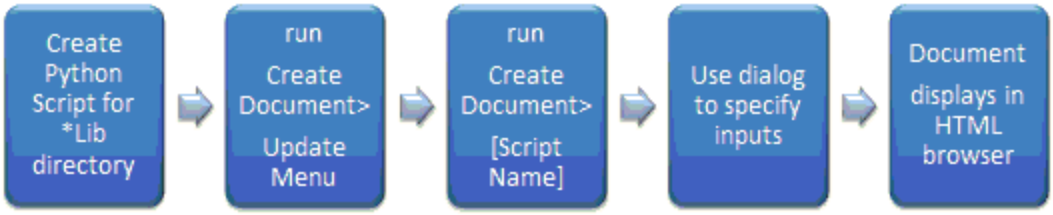
9. Click **Draw** to draw the 2D or 3D cross-section geometry. A validation check is run. This catches errors like any variable missing a value, or any parameter missing a value. For 3D designs, if you chose to **Extrude**, the 3D geometry is drawn.

User Defined Documents (UDDs)

User defined documents (UDDs) are custom reports that you define through IronPython scripts. Once placed in a Lib directory, you can access the scripts via the **Create Document** command. The scripts describe a **Create User Defined Document** dialog that lets you specify trace and solution inputs. After you confirm your input selections, an xml, html and pdf document is generated. A web browser window opens to display the generated html file. The created document appears in the Project tree, under Results in the Documents folder.



The general UDD process flow is as follows.



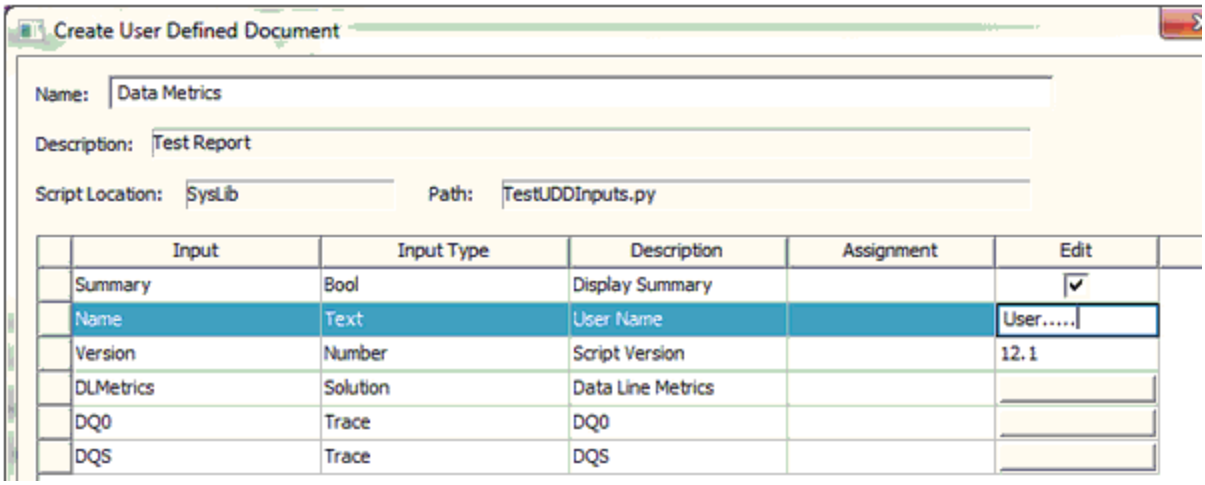
The UDD python scripts must be placed in the **UserDefinedDocuments** directory under either of **syslib**, **userlib** or **Personallib** with any subdirectory structure needed. The Lib directory can contain python scripts that have common code that other scripts can use.

Use **Results>Create Document>Update Menu** to refresh the menu to include the new UDD scripts that have been copied to syslib, userlib or Personallib, or to exclude them if they have been deleted, after the launch of desktop.

The UDD scripts that are in syslib/UserDefinedDocuments, userlib/UserDefinedDocuments or Personallib/UserDefinedDocuments become available through the **Results >Create Document** menu.

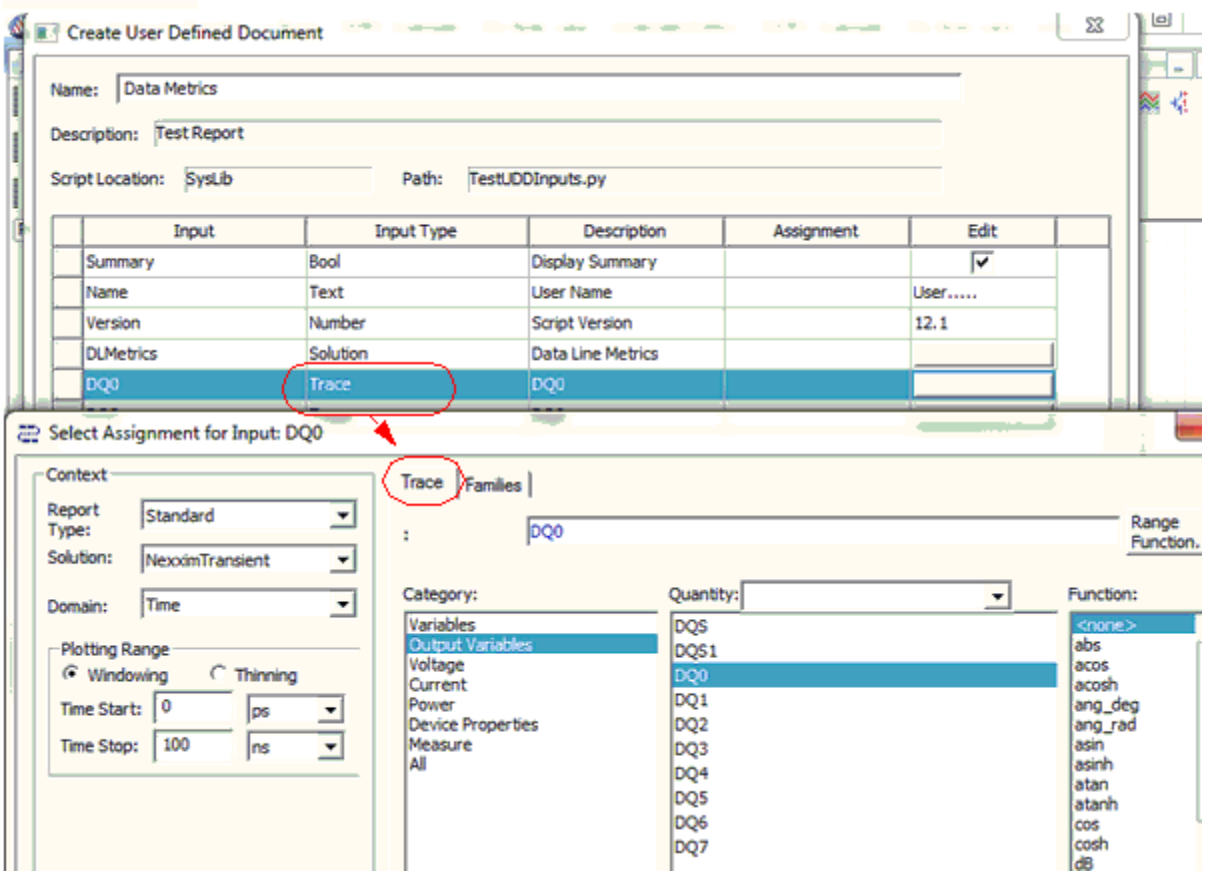
Create User Defined Document Dialog Inputs

User defined documents allow data from traces, solutions and report types as inputs. A UDD can specify the named inputs for which you select or enter the values in the **Create User Defined Document** dialog that displays when you run **Results>Create Document>**<*scriptName*>.



Input Types can be of Boolean, number, text, trace or solution type. The boolean, number and text type can be given a default value that you can interactively override when the document is created or modified. For example, you can select a trace when you create or modify a UDD document. The trace data is available to the user and can be accessed from the python script.

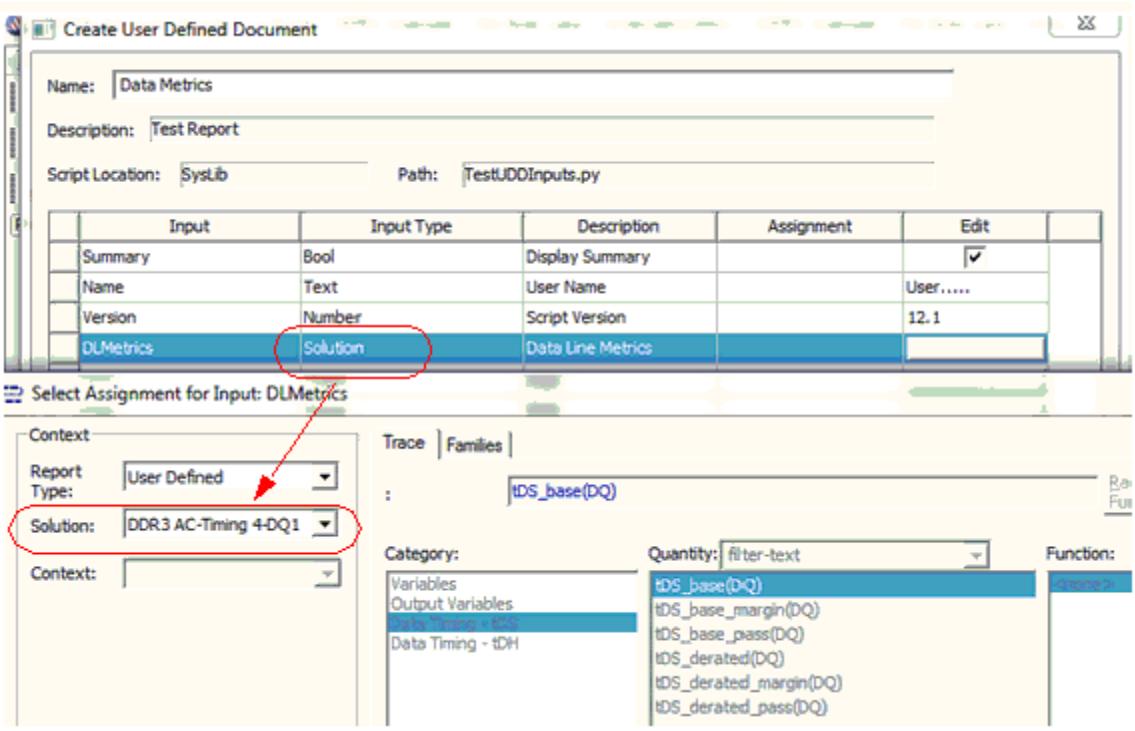
At the time of selection you can choose from the **Reporter** dialog box, the report type (Standard, Eye Diagram, User Defined), solution name, context and the quantity for which you want the trace data.



Input Type can also be Solution. You can select an entire solution when the document is created or modified. The solution data in its entirety, is now available to the user and can be accessed from the python script.

At the time of selection you can choose from the reporter dialog box, the report type (Standard, Eye Diagram, User Defined), solution name and context. A specific quantity cannot be selected since data for all quantities in the solution are available.

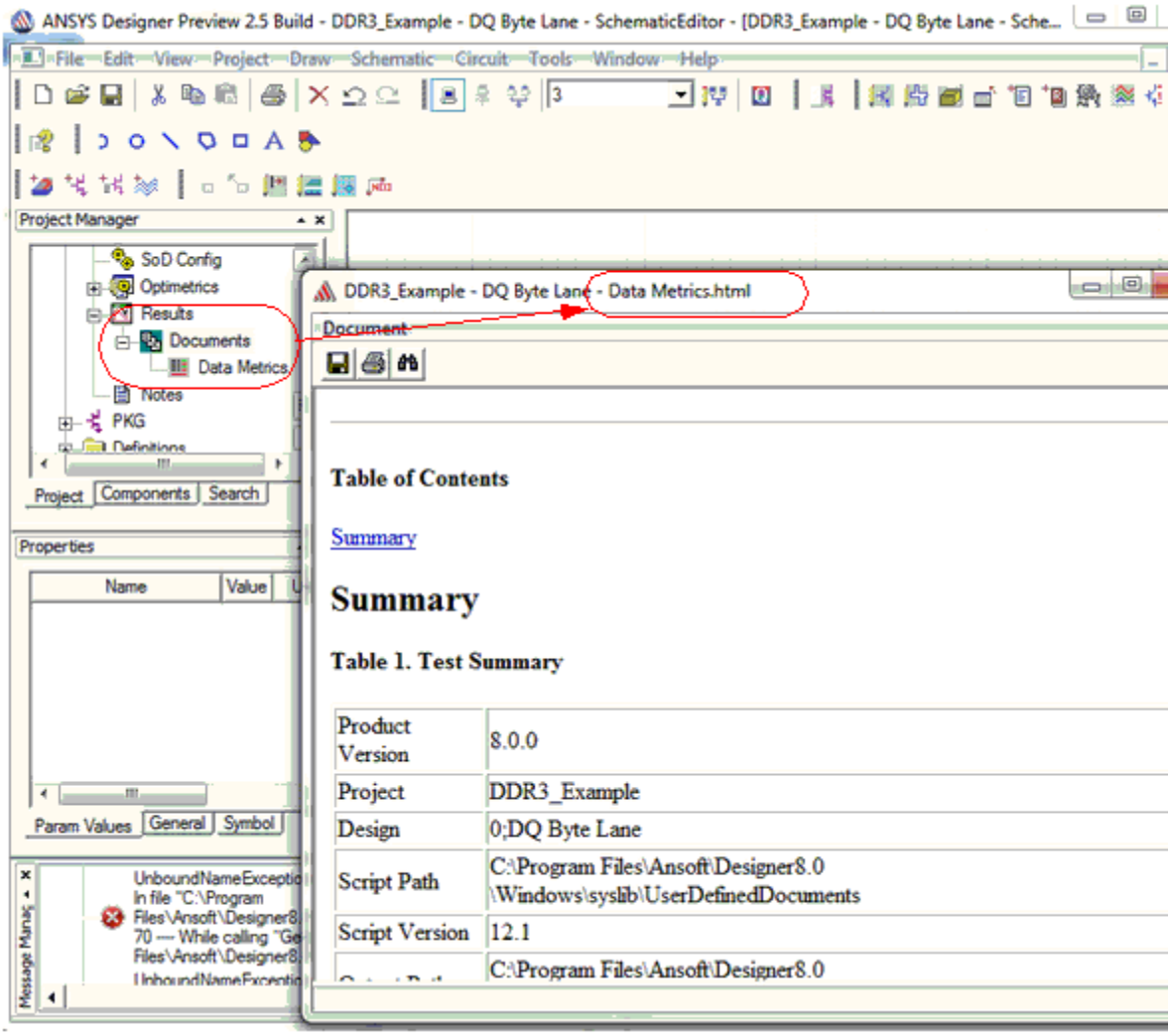
(Note: The category/Quantity/Function portion of the dialog is disabled for user input)



UDD Document Creation and Display

After all the input selections for a UDD are confirmed, based on the script, an xml, html and pdf document is generated based on the inputs provided by the user. (The xml, html and pdf generation is based on specific calls in the python script, which are explain in a following section). A web browser window also opens to display the generated html file.

The created document will be placed under a new folder named "Documents" under the "Results" folder. All documents that are created by the user for the design will be placed under this folder.



Related Topics

Managing Documents Listed in the Project Window Under Results

Viewing UDDs with an Html Web Browser

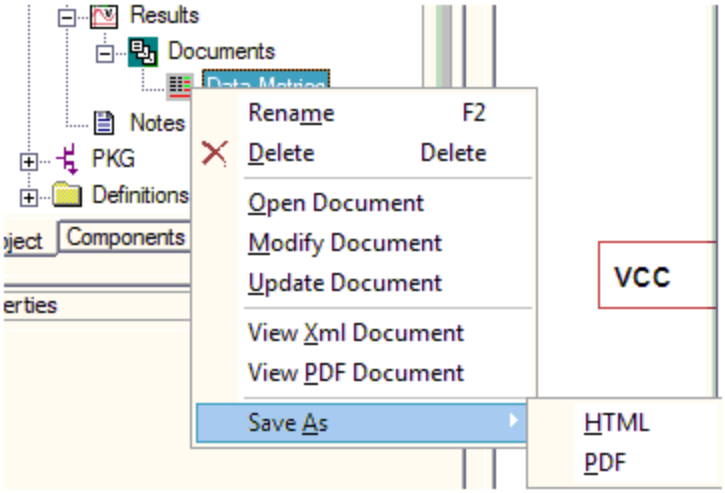
UDD Script Libraries

User Defined Documents: Python Script API

Managing Documents Listed in the Project Window Under Results

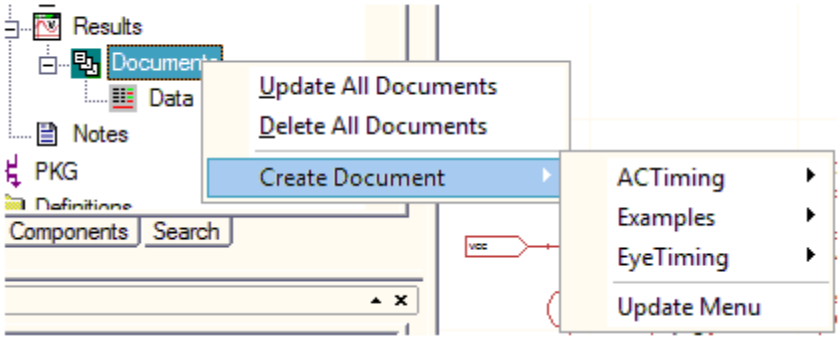
Right click on a user defined document displayed in the Project Manager tree to bring up a menu where you can rename, delete the document. **Open document** opens the web browser with the html document. **Modify document** opens the setup dialog where you can change the selections

for the input. To view the xml and the pdf document simply choose the appropriate menu items. There is also a menu item to save the document in a different location.



Documents folder right click menu

Right click on the documents folder has the menu options to Update All Documents or Delete All Documents. It also provides the option of creating a document from here.

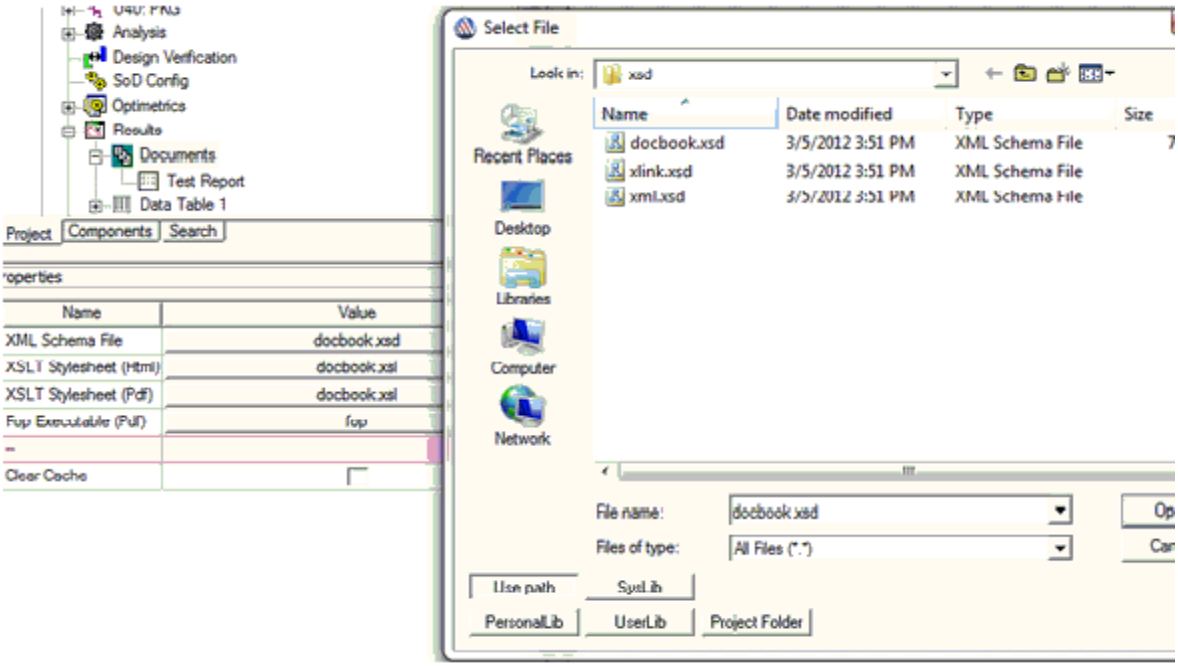


Document folder Property window

When the documents folder is selected, the Property window shows the following properties

- XML Schema File - File path to the XML schema file.
- XSLT StyleSheet (Html) - File path to the XSLT stylesheet file used for Html generation.
- XSLT StyleSheet (Fo) - File path to the XSLT stylesheet file used for Pdf generation.
- Fop Executable (Pdf) - File path the Fop executable used for Pdf generation.
- Clear Cache - Clears the cached XSL transform object and forces creation of a new one. (The caching is done to save time during document generation, so subsequent generation

or update of the document can use the cached transform object. But sometimes you may want to force a recompile of the document if you change the stylesheet).



The XML, HTML and PDF generation require the XML schema file and XSLT stylesheets to generate proper output. In addition, the PDF generation requires a FOP executable. You can use the defaults provided in the installation or provide the file paths of your own preferred stylesheets and fop executable installed in his machine.

Viewing UDDs with an Html Web Browser

The XML and HTML documents can be viewed in a web browser with some basic functionality like printing the document, searching the document for a phrase or sentence and saving the document.

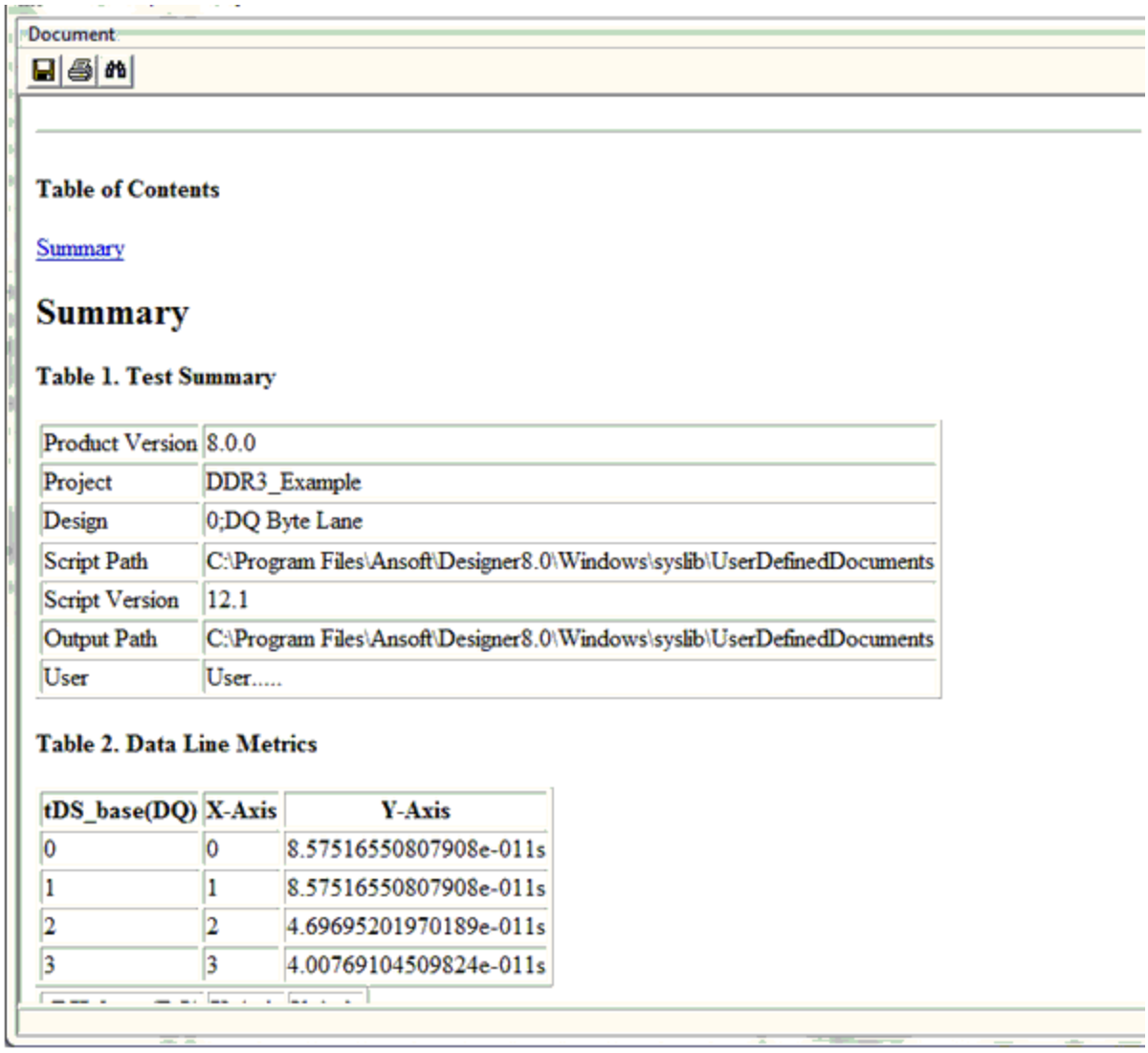

Document

Table of Contents

Summary

Summary

Table 1. Test Summary

Product Version	8.0.0
Project	DDR3_Example
Design	0;DQ Byte Lane
Script Path	C:\Program Files\Ansoft\Designer8.0\Windows\syslib\UserDefinedDocuments
Script Version	12.1
Output Path	C:\Program Files\Ansoft\Designer8.0\Windows\syslib\UserDefinedDocuments
User	User.....

Table 2. Data Line Metrics

tDS_base(DQ)	X-Axis	Y-Axis
0	0	8.57516550807908e-011s
1	1	8.57516550807908e-011s
2	2	4.69695201970189e-011s
3	3	4.00769104509824e-011s

Related Topics

UDD Script Libraries

Base classes and data files shared between similar UDDs can be organized to reuse the code in a better way. All script-library and other support files need to be in a Lib sub-directory under the UserDefinedDefinitions directory. Any .py files found in such Lib directories are ignored and not displayed in the GUI as a valid UDD choice. For a UDD script at any given directory depth, all Lib directories in its parent directories will be automatically added to the system include path (and so, any support script files from any Lib directory till the top level UserDefinedDefintions directory can be imported)

The UDD functionality uses IronPython so we have access to all the .NET assembles. If needed, any subset of the UDD functionality can be implemented in any .NET language and used by the UDD script. There are simple rules to follow to achieve this.

1. Build your .NET assembly for .NET 2.0 runtime.
2. Drop the built assembly in any Lib directory upstream of the UDD script location: that is, if you have your UDD script in C:\Users\x\PersonalLib\UserDefinedDefintions\a\b\c\myudd.py and have a .NET assembly called com.Acme.UDDLib You can keep the .NET assembly under
   - UserDefinedDefintions\Lib,
   - UserDefinedDefintions\a\Lib,
   - UserDefinedDefintions\a\b\Lib
   - UserDefinedDefintions\a\b\c\Lib
3. Add the following line to your python script
   - Import clr
   - clr.AddReference("com.Acme.UDDLib")
   - import com.Acme.UDDLib -or-- from com.Acme.UDDLib import * etc

   If for some reason you cannot place the .NET assemblies into a Lib directory under UserDefinedDefintions, you need to do a couple more steps before step 3 listed above.

```
Import sys
sys.path.append("full path to your .NET assembly location")
```

Related Topics

User Defined Documents: Python Script API

User Defined Documents: Python Script API

A User Defined Documents (UDD) extension is implemented as an IronPython script that defines a class with a specific name: **UDDExtension** which derives from a specific base class **IUDDPluginExtension** and implements its abstract methods.

Import Statements

The base class to be used and the types it uses in turn are contained in .NET assemblies. The use of these requires that the assemblies be imported into the UDD script: the following import statements should be added to the top of the python script:

```
from
Ansys.Ansoft.DocGeneratorPluginDotNet.DocGenerator.API.Data
import *
from
Ansys.Ansoft.DocGeneratorPluginDotNet.DocGenerator.API.Interfac
es import *
```

UDDExtension Class

The UDD itself should be implemented as an IronPython class called **UDDExtension** which must derive from the **IUDDPluginExtension** abstract base class (from the **Ansys.Ansoft.DocGeneratorPluginDotNet.DocGenerator.API.Interfaces** namespace).

Note that power users could derive a class hierarchy tuned toward a specific type of UDDs and that they can derive from their own base classes. The only requirement is that directly or indirectly, the UDD class must derive from **IUDDPluginExtension**.

Example:

```
def BaseClassUDD ((IUDDPluginExtension):
#base class implementation
…
def UDDExtension ((BaseClassUDD):
#UDD class implementation
…
```

Note	All of the above text has been copied from the help section for the UDOs and modified for the UDDs. Since the UDDs are modelled after the UDOs, the usage is also similar.

IUDDPluginExtension Abstract Class

Required functions:

The **IUDDPluginExtension** abstract class declares the following abstract methods that must be implemented in the UDDExtension class or one of its base classes. Not implementing any of these methods will result in a run-time error and a non functioning UDD.

GetUDDName() : Return a string that is used as a prefix for all solution instances created using this UDD.

Example:

```
def GetUDDName(self):
return "MinMaxAvg
```

GetUDDDescription() : Returns a description for the UDD, its purpose etc.

Example:

```
def GetUDDDescription(self):
return "Sample UDD"
```

ShowDefaultSetupDialog() : Returns True if the default dialog is to be shown. Return False if the user does not want the default dialog box. In this case the user might want to implement/show a customized setup dialog.

Example:

```
def ShowDefaultSetupDialog(self):
return True
```

GetUDDInputParams(List<UDDInputParams> uddInputs) : Returns the list of inputs parameters for the User Defined Document. Returns boolean: True on success, False on failure.

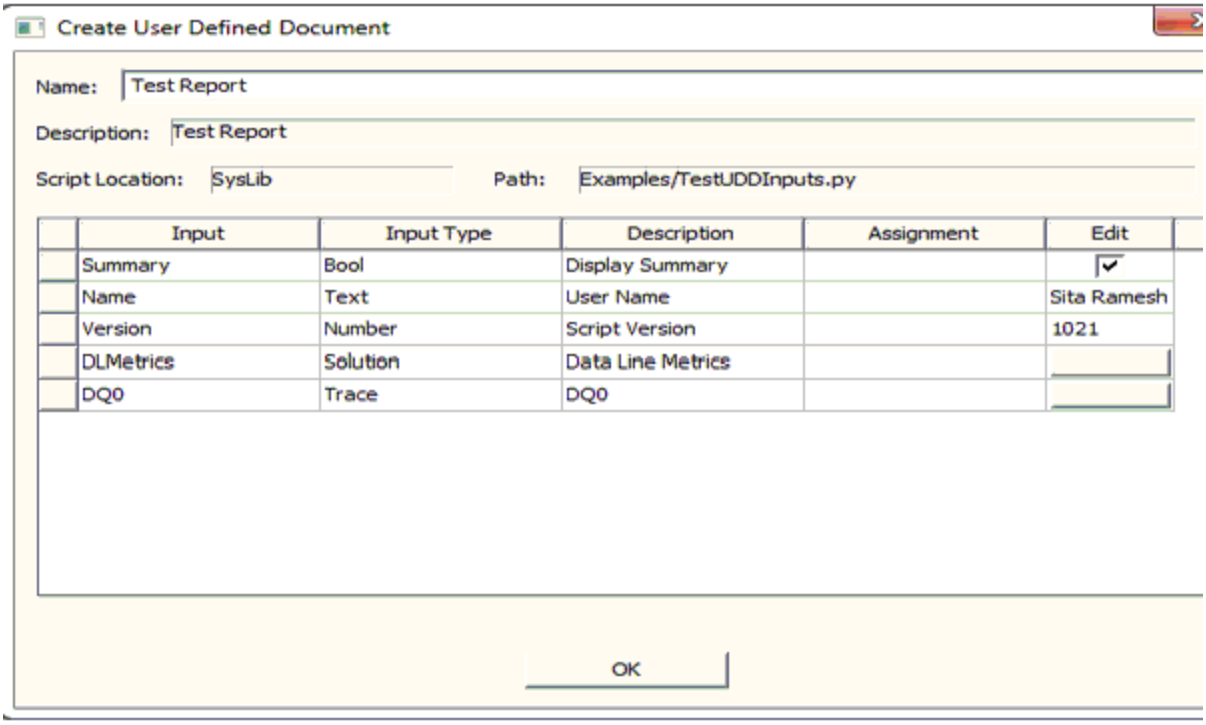
The supplied input parameters are used to populate details of the parameters to which the UDD user will specify value, specify the input names and their types.

uddInputs: .NET list of UDDInputParams objects. The UDD script is expected to add one instance of UDDInputParams for each input definition it wants displayed. The UDD user will, when creating the UDD, assign a matching value to each such input.

Example:

```
def GetUDDInputParams(self, uddInputs)
# Boolean input
param1 = UDDInputParams("Summary","Display Summary",
Constants.kBoolTypeStr, True)
uddInputs.Add(param1)
# Text input
param2 = UDDInputParams("Name","User Name",
Constants.kTextTypeStr , "Sita Ramesh")
uddInputs.Add(param2)
# Number input
param3 = UDDInputParams("Version","Script Version",
Constants.kNumberTypeStr, 1021)
uddInputs.Add(param3)
# Solution input
param5 = UDDInputParams("DLMetrics","Data Line
Metrics",Constants.kSolutionTypeStr)
uddInputs.Add(param4)
# Trace input
param5 = UDDInputParams("DQ0","DQ0",Constants.kTraceTypeStr)
uddInputs.Add(param5)

return True
```

Based on the input params the following dialog is displayed when you click **Reports>Create Document**. The name and description of the UDD are also displayed in this dialog.



Generate(List<UDDInputData> uddInputs, IUDDGenerator generator, IProgressMonitor progressMonitor) : This is the main method which accesses the data from the uddInputs and generates the document.

uddInputs: The list of inputs that the user setup in the dialog box. They are now available to query for data.

generator: This is the document generator object which we use to create different elements of the document like titles, sections, tables, images and write the data too. This interface is explained in the Document Generator Interface document.

progressMonitor : IProgressMonitor object. This can be used to set progress for long running calculations, check for user initiated abort etc.

Example:

```
def Generate(self, input, docgen, progMon):

# Gather data from inputs
boolinput = input[0].Data()
textinput = input[1].Data()
dblinput = input[2].Data()

# Get document root
```

```
docroot = docgen.GetDocumentRoot()

# Add Section
section1 = docroot.AddSection("Summary", "Overall Results ")

# Add a table
table1 = section1.AddTable("Test Summary")

#Add a table group with 2 columns
tgroup1 = table1.AddTableGroup(2)

# get desktop application
oApp = self.GetUDDAppContext()
if oApp != None:
oDesktop = oApp.GetAppDesktop()
if oDesktop != None:
# version number
version = oDesktop.GetVersion()
text1 = tgroup1.AddContent()
text1 .Add(0, "Product Version")
text1 .Add(1, version)

oProject = oDesktop.GetActiveProject()
if oProject != None:
projectname= oProject.GetName()
text1 = tgroup1.AddContent()
text1 .Add(0, "Project")
text1 1.Add(1, projectname)

oDesign = self.GetUDDDesignContext()
if oDesign != None:
designname = oDesign.GetName()
text1 = tgroup1.AddContent()
text1 .Add(0, "Design")
text1 .Add(1, designname)

# Provides a script path
```

```
scriptpath = docgen.GetScriptPath()
text1 = tgroup1.AddContent()
text1 .Add(0, "Script Path")
text1 .Add(1, scriptpath )

#Provides the script version
text1 = tgroup1.AddContent()
text1 .Add(0, "Script Version")
text1 .Add(1, str(dblinput ))

#Provides the output xml path
outputpath = docgen.GetOutputFilePath()
text1 = tgroup1.AddContent()
text1 .Add(0, "Output Path")
text1 .Add(1, outputpath )

#Provides the user information
text1 = tgroup1.AddContent()
text1 .Add(0, "User")
text1 .Add(1, textinput)

# Generate Xml output
docgen.Write(False)

# Generate Html output
docgen.WriteHTML()

# Generate PDF output
docgen.WritePDF()

return True
```

Optional functions:

SetupUDDInputParams(List<UDDInputParams> uddInputs) : Displays a customized dialog and returns the user choices for the input params.

uddInputs- .NET list of UDDInputParams objects with values for each of them. These can be the user choice for each input obtained through a custom dialog or some other non graphical assignment.

We cannot process trace and solution types of input with a custom dialog because there is no way of assigning solution data to the input without the invocation of the reporter dialog.

Example:

```
    def SetupUDDInputParams(self, uddInputs)
    udddialog = BaseExampleUDDDialog()
    if udddialog.ShowDialog() == Forms.DialogResult.OK:
# Boolean input
param1 = udddialog.GetInput("Summary")
uddInputs.Add(param1)

# Text input
param2 = udddialog.GetInput("Name")
uddInputs.Add(param2)

# Number input
param3 = udddialog.GetInput("Version")
uddInputs.Add(param3)
```

HandleUDDEvents(List<string> eventTags) : The tags associated with the event is received by plugin using this abstract class.

This method is the event handler for all link events set by the SetEventLink() method on a IUDDText. Refer to the definition of the IUDDText object in the Document Generator Interface document.

Example:

```
    def HandleUDDEvents(self, uddLinks):
    if uddLinks[0] == "Open" Report":
# Get Design Name
oDesign = self.GetUDDDesignContext()
if oDesign != None:
  oDesign.OpenReport(uddLinks[1])
    return True
```

GetUDDSchema() : Returns the file path of the schema to validate the xml. This will override the default schema used. Return string containing the full file path of the schema.

```
    def GetUDDSchema(self):
```

```
return "C:\\Program
Files\\Ansoft\\Designer8.0\\Windows\\common\\docbook\\schema\\x
sd\\docbook.xsd"
```

GetUDDStyleSheetForHtml() : Returns the file path of the style sheet used to generate the html document. This will override the default stylesheet for html. Returns string containing the full file path of the style sheet.

```
def GetUDDStyleSheetForHtml(self):
return "C:\\Program
Files\\Ansoft\\Designer8.0\\Windows\\common\\docbook\\"
```

GetUDDStyleSheetForPdf() : Returns the file path of the style sheet used to generate the pdf document. This will override the default stylesheet for pdf. Returns string containing the full file path of the style sheet.

Example:

```
def GetUDDStyleSheetForPdf(self):
return "C:\\Program
Files\\Ansoft\\Designer8.0\\Windows\\common\\docbook\\xsl\\fo\\
docbook.xsl"
```

GetFopExecutable() : Returns the file path of the fop executable used to generate the pdf document. This will override the default stylesheet for pdf.Returns string containing the full file path of the fop executable.

Example:

```
def GetFopExecutable(self):
return "C:\\Program
Files\\Ansoft\\Designer8.0\\Windows\\common\\ApacheFOP\\fop-
1.0\\fop"
```

GetUDDAppContext() : Returns the UDD Owner (if set). This is a Dispatch wrapper that is essentially a COM IDispatch implementation and corresponds to the IDispatch pointing to the desktop app.

GetUDDDesignContext() : Returns the UDD Owner (if set). This is a Dispatch wrapper that is essentially a COM IDispatch implementation and corresponds to the IDispatch pointing to the Design.

Data Types Used in Python Script

There are several types that you must use while authoring the python script. Some of them are used to pass data from UI to python script and to provide interface for working with this data. Some are used to pass data from python script to UI.

To pass data from python script to UI the objects of the C# class must be created in python script using their C# constructors. Then they can be set as functions return values or set to the output parameters using their API.

Constants class

kTraceTypeStr : string constant used to specify an input of trace type

kSolutionTypeStr : string constant used to specify an input of solution type

kNumberTypeStr : string constant used to specify an input of number type

kTextTypeStr : string constant used to specify an input of text type

kBoolTypeStr : string constant used to specify an input of boolean type

kStandardReportStr : string constant to specify a standard report

kEyeDiagramReportStr : string constant to specify an eye diagram report

kUserDefinedReportStr : string constant to specify a user defined report

kSweepDomainStr : string constant to specify the sweep domain

kTimeDomainStr : string constant to specify the time domain

UDDInputParams class

The objects of this class must be created in python script in the **GetUDDInputParams()** function and the **SetUDDInputParams()** function.

Attributes :

Input Name (string)

Input Description (string)

Input Type (Can be Boolean, Number, Text, Trace or Solution) (string)

BoolData (boolean)

DoubleData (double)

TextData (string)

ReportType (string)

SolutionName (string)

DomainName (string)

Constructors:

UDDInputParams(string name, string description, string type)

UDDInputParams(string name, string description, string type, bool data)

UDDInputParams(string name, string description, string type, double data)

UDDInputParams(string name, string description, string type, string data)

UDDInputParams(string name, string description, string type, string reportType, string solutionName, string domainName)

Property Accessors :

Name : Get/Set the name of an input

Description : Get/Set the desccription of an input

Type : Get/Set the type of an input

BoolData : Get/Set the data of a boolean input

DoubleData : Get/Set the data of a number input

TextData : Get/Set the data of a text input

ReportType : Get/Set the report type

SolutionName : Get/Set the name of the solution

DomainName : Get/Set the name of the domain

IProgressMonitor Abstract Class

The object of this class is a progress monitor. It is used to display calculations progress in UI and check is the user has requested an abort of the computation.

When displayed in the application, each progress message has four items:

A task name

A sub-task name

The progress amount

A button to abort the task in progress.

All of this functionality and abort interaction is achieved using the following functions.

SetTaskName (string taskName):

SetSubTaskName (string subTaskName)

BeginTask (string name)

SetTaskProgressPercentage(int progressPercent)

CheckForAbort(): If the quantities being generated are computationally expensive, the UDO author can periodically call this method and then call EndTask with Fail and return False.

EndTask (bool passFail)

Example:

```
progMon.BeginTask("Process DQS")
progMon.SetSubTaskName("Compute UI segments")
progMon.SetTaskProgressPercentage(33)
progMon.SetSubTaskName("Compute the rest")
progMon.SetTaskProgressPercentage(100)
progMon.EndTask(True)
```

UDD Input interfaces

The Generate function takes in a list of inputs. These input interfaces allow the user to access data from the design.

IUDDInputBool : This interface exposes 3 methods

Name() : Gets the inputs name.

Type() : Gets the input type.

Data() : Gets the boolean data, set by the user in the setup dialog.

IUDDInputDouble : This interface exposes 3 methods

Name() : Gets the inputs name.

Type() : Gets the input type.

Data() : Gets the double data, set by the user in the setup dialog.

IUDDInputText : This interface exposes 3 methods

Name() : Gets the inputs name.

Type() : Gets the input type.

Data() : Gets the text data, set by the user in the setup dialog.

IUDDInputTrace : This interface exposes 3 methods

Name() : Gets the inputs name.

Type() : Gets the input type.

DoubleData() : Method used to return x and y double data as a IDictionary<double, double>

DoubleData(IDictionary<string, string> variation) : Method used to return x and y double data as a IDictionary<double, double>, given a variation.

ComplexData() : Method used to return x data and y complex data as a IDictionary<double, double[]>

ComplexData(IDictionary<string, string> variation) : Method used to return x data and y complex data as a IDictionary<double, double[]>, given a variation.

TextData() : Method used to return x data and y data as a IDictionary<string, string>

TextData(IDictionary<string, string> variation) : Method used to return x data and y data as a IDictionary<string, string>, given a variation.

VariableValues() : Method used to get a list of variations as a IList<Dictionary<string, string>>

IUDDInputSolution : This interface exposes 3 methods

Name() : Gets the inputs name.

Type() : Gets the input type.

DoubleData(string name) : Method used to return x and y double data as a IDictionary<double, double>, given a quantity name.

DoubleData(string name, IDictionary<string, string> variation) : Method used to return x and y double data as a IDictionary<double, double>, given a quantity name and a variation.

ComplexData(string name) : Method used to return x data and y complex data as a IDictionary<double, double[]>, given a quantity name.

ComplexData(string name, IDictionary<string, string> variation) : Method used to return x data and y complex data as a IDictionary<double, double[]>, given a quantity name and a variation.

TextData(string name) : Method used to return x data and y data as a IDictionary<string, string> given a quantity name.

TextData(string name, IDictionary<string, string> variation) : Method used to return x data and y data as a IDictionary<string, string>, given a quantity name and a variation.

CategoryNames() : Method to return a list of category names in the solution as an IList<string>

QuantityNames(string category) : Method to return a list of quantity names in the solution as an IList<string>, given a category.

VariableValues() : Method used to get a list of variations as a IList<Dictionary<string, string>>

Examples:

```
def Generate(self, input, docgen, progMon):
# Getting the boolean data set by the user
boolinput = input[0].Data()
# Getting the double data set by the user
dblinput = input[1].Data()
# Getting the text data set by the user
textinput = input[2].Data()
# Getting the category names in a solution
categories = input[3].CategoryNames()
# Getting the quantity names based on a category
```

```
quantities = input[3].QuantityNames(categories[0])
# Getting the XY data from the trace
xydata = input[4].DoubleData()
```

User Defined Document Scripting Interface

To access the UserDefineddocuments scripting object, use:

```
Set oModule = oDesign.GetModule("UserDefinedDocuments")
```

Once you have the scripting object, you can use the following methods:-

1. `AddDocument([in] VARIANT data, [in] VARIANT traces, [out, retval] BSTR* uniqueName)`
   a. Takes a VARIANT data which defines the document.
   b. Takes a VARIANT trace data for the inputs in the document.
   c. Returns a unique name
2. `EditDocument( [in] BSTR originalName, [in] VARIANT modifiedData, [in] VARIANT modifiedTraces, [out, retval] BSTR* uniqueName)`
   a. Takes the name of the original document.
   b. Takes a VARIANT data which defines the edited document.
   c. Takes a VARIANT trace data for the inputs in the document.
   d. Returns a unique name
3. `RenameDocument( [in] BSTR oldName, [in] BSTR newName)`
   a. Takes the name of the original document.
   b. Takes the new name of the document.
4. `DeleteDocument( [in] BSTR name)`
   a. Takes the name of the document to be deleted.
5. `UpdateDocument( [in] BSTR name)`
   a. Takes the name of the document to be updated.
6. `ViewHtmlDocument( [in] BSTR name)`
   a. Takes the name of the document to be viewed in HTML.
7. `ViewPdfDocument( [in] BSTR name)`
   b. Takes the name of the document to be viewed as a PDF.
8. `SaveHtmlDocumentAs( [in] BSTR name, [in] BSTR saveTo)`
   a. Takes the name of the document to be saved.
   b. Takes the file path to save the socument as.
9. `SavePdfDocumentAs( [in] BSTR name, [in] BSTR saveTo)`
   a. Takes the name of the document to be saved.
   b. Takes the file path to save the socument as.

10. `GetDocumentDefinitionNames( [in] BSTR separator, [out, retval] VARIANT* names)`
    a. 'separator' is used to convey the directory "level"
    b. Returns the (file) names of doc definitions according to the files in various installation directories.
11. `DeleteAllDocuments()`
12. `UpdateAllDocuments()`

For 6, 7, 8, and 9, the document must have an existing, generated HTML or PDF.

The UserDefinedDocument Data format in the script:

To define a document in VB script:

Array("NAME:Test Report", (Name of the document)
"Test Report", (Description of the document)
"SysLib", (Location of the python script(Syslib, Userlib, PeronalLib etc)
"TestUDDReport", (Relative path of the script in the UserDefinedDocuments folder)

// Start of input definition //
Array("NAME:Inputs", (Document Inputs keyword)
// Solution input //
Array("NAME:DLMetrics", (Input name)
"Solution", (Solution Input Type)
"Data Line Metrics", (Input Description)
-1, (Solution ID)
-1), (Report ID)
// Trace input //
Array("NAME:DQ0", (Input name)
"Trace", (Trace Input Type)
"DQ0", (Input Description)
-1, (Solution ID)
-1), (Report ID)
// Text input //
Array("NAME:Name", (Input name)
"Text", (Text Input Type)
"User Name", (Input Description)
Array("Sita Ramesh")), (Default Value)
// Bool input //

Array("NAME:Summary", (Input name)
"Bool", (Boolean Input Type)
"Display Summary", (Input Description)
Array(true)), (Default Value)
// Number input //
Array("NAME:Version", (Input name)
"Number", (Number Input Type)
"Script Version", (Input Description)
Array(1021)))), (Default Value)
// Trace selection for the solution and trace inputs //
Array("NAME:DocTraces", (Document traces keyword)
// For input "DLMetrics" //
Array("NAME:DLMetrics", (Input name)
// Trace defintion similar to the UDO. This trace definition is a User defined solution //
Array("User Defined",
"", "DDR3 AC-Timing 4-DQ1", Array("Context:=", ""), Array("Index:=", Array("All"), "Trise:=", Array ("Nominal"), "Tfall:=", Array("Nominal"), "Pulse_Width:=", Array("Nominal"), "Data_Rate:=", Array ("Nominal"), "Length:=", Array("Nominal")), Array("Probe Component:=", Array("")), Array())),
// For input "DQ0" //
Array("NAME:DQ0",
// Trace defintion similar to the UDO. This trace definiton is a Standard solution //
Array("Standard", "DQ0", "NexximTransient", Array("NAME:Context", "SimValueContext:=", Array(1, 0, 2, 0, false, false, -1, 1, 0, 1, 1, "", 0, 0, "DE", false, "0", "DP", _
false, "20000000", "DT", false, "0.001", "WE", false, "100ns", "WM", false, _
"100ns", "WN", false, "0ps", "WS", false, "0ps")), Array("Time:=", Array("All"), "Trise:=", Array(_
"Nominal"), "Tfall:=", Array("Nominal"), "Pulse_Width:=", Array("Nominal"), "Data_Rate:=", Array ("Nominal"), "Length:=", Array("Nominal")), Array("Probe Component:=", Array(_
"DQ0")), Array())))

Python Script to Define Document

To define a document in Python script:

```
[
"NAME:Test Report", "Test Report", "SysLib",
"Examples/TestUDDInputs",
[
"NAME:Inputs",
```

```
[
"NAME:DLMetrics","Solution", "Data Line Metrics", -1, -1
],
[
"NAME:DQ0", "Trace", "DQ0", -1, -1
],
[
"NAME:DQS", "Trace", "DQS", -1, -1
],
[
"NAME:Name", "Text", "User Name", ["Sita Ramesh"]
],
[
"NAME:Summary", "Bool", "Display Summary", [True]
],
[
"NAME:Version", "Number", "Script Version" [1021]
]
]
],
[
"NAME:DocTraces",
[
"NAME:DLMetrics",
[
"User Defined", "", "DDR3 AC-Timing 4-DQ1",
[
"Context:=" , ""
],
[
"Index:=" , ["All"], "Trise:=" , ["Nominal"], "Tfall:=" ,
["Nominal"], "Pulse_Width:=" ,
["Nominal"], "Data_Rate:=" , ["Nominal"], "Length:=" ,
["Nominal"]
],
[
"Probe Component:=" , [""]
```

```
],
[]
]
],
[
"NAME:DQ0",
[
"Standard", "DQ0", "NexximTransient",
[
"NAME:Context", "SimValueContext:=" ,
[1,0,2,0,False,False,-
1,1,0,1,1,"",0,0,"DE",False,"0","DP",False,"20000000","DT",Fals
e,"0.001","WE",False,"100ns","WM",False,"100ns","WN",False,"0ps
","WS",False,"0ps"]
],
[
"Time:=" , ["All"],"Trise:=" , ["Nominal"],"Tfall:=" ,
["Nominal"],"Pulse_Width:=" ,["Nominal"],
"Data_Rate:=" , ["Nominal"], "Length:=" , ["Nominal"]
],
[
"Probe Component:=" , ["DQ0"]
],
[]
]
],
]
```

Sample Script:- This one adds, edits, renames and deletes a document

```
Set oModule = oDesign.GetModule("UserDefinedDocuments")
' Add a UDD
oModule.AddDocument Array("NAME:Test Report1", "Test Report",
"SysLib", _
"Examples/TestUDDInputs", Array("NAME:Inputs", Array
("NAME:DLMetrics", "Solution", _
```

```
"Data Line Metrics", -1, -1), Array("NAME:DQ0", "Trace", "DQ0",
-1, -1), Array("NAME:DQS",
"Trace", "DQS", -1, -1), Array("NAME:Name", "Text", "User
Name", Array("Sita Ramesh")), Array("NAME:Summary", "Bool",
"Display Summary", Array(true)), Array("NAME:Version",
"Number", "Script Version"))), Array("NAME:DocTraces", Array
("NAME:DLMetrics", Array("User Defined", "", "DDR3 AC-Timing 4-
DQ1", Array("Context:=", ""), Array("Index:=", Array("All"),
"Trise:=", Array( "Nominal"), "Tfall:=", Array("Nominal"),
"Pulse_Width:=", Array("Nominal"), "Data_Rate:=", Array(
"Nominal"), "Length:=", Array("Nominal")), Array("Probe
Component:=", Array("")), Array())), Array("NAME:DQ0", Array( _
"Standard", "DQ0", "NexximTransient", Array("NAME:Context",
"SimValueContext:=", Array( _
1, 0, 2, 0, false, false, -1, 1, 0, 1, 1, "", 0, 0, "DE",
false, "0", "DP", false, "20000000", "DT", false, "0.001",
"WE", false, "100ns", "WM", false, "100ns", "WN", false, "0ps",
"WS", false, "0ps")), Array("Time:=", Array("All"), "Trise:=",
Array("Nominal"), "Tfall:=", Array("Nominal"), "Pulse_Width:=",
Array("Nominal"), "Data_Rate:=", Array("Nominal"), "Length:=",
Array("Nominal")), Array("Probe Component:=", Array("DQ0")),
Array())))

' Edit Document
oModule.EditDocument "Test Report1", Array("NAME:Test Report",
"Test Report", _
"SysLib", "Examples/TestUDDInputs", Array("NAME:Inputs", Array
("NAME:DLMetrics", _
"Solution", "Data Line Metrics", 1000001, 0), Array("NAME:DQ0",
 "Trace", "DQ0", 32, _
2), Array("NAME:DQS", "Trace", "DQS", 32, 4), Array
("NAME:Name", "Text", "User Name", Array( "Sita Ramesh")),
Array("NAME:Summary", "Bool", "Display Summary", Array(true)),
Array("NAME:Version", "Number", "Script Version"))), Array
("NAME:DocTraces", Array("NAME:DLMetrics", Array( "User
Defined", "Solution", "DDR3 AC-Timing 4-DQ1", Array
("Context:=", ""), Array("Index:=", Array( "All"), "Trise:=",
Array("Nominal"), "Tfall:=", Array("Nominal"), "Pulse_Width:=",
Array( "Nominal"), "Data_Rate:=", Array("Nominal"), "Length:=",
Array("Nominal")), Array("Probe Component:=", Array("")), Array
())), Array("NAME:DQ0", Array("Standard", "DQ1",
"NexximTransient", Array("NAME:Context", "SimValueContext:=",
```

```
Array(1, 0, 2, 0, false, false, -1, 1, 0, 1, 1, "", 0, 0, "DE",
false, "0", "DP", _
false, "20000000", "DT", false, "0.001", "WE", false, "100ns",
"WM", false, "100ns", "WN", false, "0ps", "WS", false, "0ps")),
 Array("Time:=", Array("All"), "Trise:=", Array("Nominal"),
"Tfall:=", Array("Nominal"), "Pulse_Width:=", Array("Nominal"),
"Data_Rate:=", Array("Nominal"), "Length:=", Array("Nominal")),
Array("Probe Component:=", Array("DQ1")), Array())))

' Rename a UDD
oModule.RenameDocument "Test Report", "Test UDD Report"

' Update UDD
oModule.UpdateDocument "Test UDD Report"

' View Html
oModule.ViewHtmlDocument "Test UDD Report"

' View Pdf
oModule.ViewPdfDocument "Test UDD Report"

' Save Html
oModule.SaveHtmlDocumentAs "Test UDD Report",
"c:/AnsysProjects/Test.html"

' Save pdf
oModule.SavePdfDocumentAs "Test UDD Report",
"c:/AnsysProjects/Test.pdf"

' Delete UDD
oModule.DeleteDocument "Test UDD Report"
```

Note	The product has to implement the GetModule call to create the UserDefinedDocument scripting object. For e.g. Check AltraSimDesign.cpp (function GetMgrIDispatch()).

Document Generator Interfaces

This document briefly describes the API interfaces available in the document generator plugin. (Ansys.Ansoft.DocGeneratorPluginDotNet.dll)

Scripting objects available in the script for the Generate function

- oApp = self.GetUDDAppContext(
  Gets the application context
  Usage:- Gets the active project ad the version of the product

```
oDesktop = oApp.GetAppDesktop()
```

if oDesktop != None:
vr = oDesktop.GetVersion()
oProject = oDesktop.GetActiveProject()

- oDesign = self.GetUDDDesignContext()
  Gets the design context
  Usage:- Gets the design name.

oDesign = self.GetUDDDesignContext()
if oDesign != None:
nm = oDesign.GetName()

- IUDDGenerator interface
  This interface available in the Generate method of the UDDPluginExtension.
  This interface can be used to

1. Set the document output file path.

```
docgen.SetOutput("C:\\Examples\\DocumentOutput.xml")
```

2. Get the document root.

```
docroot = docgen.GetDocumentRoot()
```

3. Write out to the output file.

```
docgen.Write()
```

4. Write Html document

```
void WriteHTML();
```

5. Write PDF document

```
void WritePDF();
```

6. Load the Html transform object

```
void LoadHTMLTransform();
```

7. Load the cached PDF transform object

```
void LoadPDFTransform();
```

8. Get script path
```
string GetScriptPath();
```

9. Get output file path
```
string GetOutputFilePath();
```

- IUDDRoot interface

  Calling GetDocumentRoot() on the IUDDGenerator interface provides you with the this interface. This interface can be used to

1. Add a new section to the document. Provide a section title.
```
section1 = docroot.AddSection("Section title")
```

2. Add a new section to the document. Provide a section title and subtitle
```
section1 = docroot.AddSection("Section title", "Section subtitle")
```

3. Add a new title
```
section1 = docroot.AddTitle("Title")
```

4. Add a new subtitle
```
section1 = docroot.AddSubtitle("Subtitle")
```

- IUDDSection interface

  Calling AddSection() on the IUDDRoot interface provides you with the this interface. This interface can be used to

1. Set an ID for the section for internal links.
```
section1.SetID("id")
```

2. Add a new table to the document. Provide a table title.
```
table1 = section1.AddTable("Table title")
```

3. Add a new image to the document. Provide an image title and a file path to the image file.
```
image1 = section1.AddImage("Image title")
```

4. Add text to the document.
```
text1 = section1.AddText("Random text........")
```

- IUDDImage interface

  Calling AddImage() on the IUDDSection interface provides you with the this interface. On this interface you can call the following methods

1. Set an ID for the image for internal links.

   ```
   image1.SetID("id")
   ```

2. Set alignment information . can be "center", "left" and "right".

   ```
   image1.SetAligment("center")
   ```

3. Set the file path of the image file. Not necessary if image file path is set through the AddImage() method

   ```
   image1.SetFileRef("Image path")
   ```

4. Set the format of the image file. Can be "BMP", "PNG", "JPEG", "JPG", "DVI" etc.. Not sure about the necessity of this one.

   ```
   image1.SetFormat("format")
   ```

- IUDDText interface

  Calling AddText() on the IUDDSection interface provides you with the this interface. On this interface you can call the following methods

1. Set an ID for the text for internal links.

   ```
   text1.SetID("id")
   ```

2. Set the emphasis attribute on the text

   ```
   text1.SetEmphasis()
   ```

3. Set the quotes attribute on the text

   ```
   Text1.SetQuotes()
   ```

4. Set the block quotes attribute on the text

   ```
   text1.SetBlockquotes()
   ```

5. Set quotes on the text

   ```
   Text1.SetQuotes()
   ```

6. Set the wordsize attribute on the text

   ```
   text1.SetSize(size as an integer)
   ```

7. Set a link to an ID of any element to provide internal links
   ```
   text1.SetLink("linkname")
   ```

8. Set an event link to handle an event. The HandleUDDEvents method should be implemented in the script to handle the event.
   ```
   text1.SetEventLink("linkname")
   ```

- IUDDTable interface

  Calling AddTable() on the IUDDSection interface provides you with the this interface. On this interface you can call the following methods

1. Set an ID for the table for internal links.
   ```
   table1.SetID("id")
   ```

2. Set alignment information . can be "center", "left" and "right".
   ```
   table1.SetAlignment("center")
   ```

3. Set the background color of the table
   ```
   table1.SetBgColor(string bgcolor)
   ```

4. Set the frame type. Can be "all", "bottom", "top,", "sides", "topbot"
   ```
   table1.SetFrame(string frame)
   ```

5. Add a table group and specify the number of columns. A table can have multiple table groups.
   ```
   IUDDTableGroup table1.SetTableGroup(int columns)
   ```

- IUDDTableGroup interface

  Calling AddTableGroup() on the IUDDTable interface provides you with the this interface. On this interface you can call the following methods

1. Set an ID for the table group for internal links.
   ```
   tgroup1.SetID("id")
   ```

2. Set alignment information . can be "center", "left" and "right".
   ```
   tgroup1.SetAlignment("center")
   ```

3. Set the column width of a column given the index of the column and the required width. Width can be set in 2 ways.
   - Width can be set relative to 1. E.g Setting it to "2*" makes the column width double the width of the others.

- If the entire table width is considered to be 99.99 units. Width can be a number relative to this.

```
tgroup1.SetColumnWidth(int index, string width)
```

4. Add a header to the table group

```
IUDDTableRow tgroup1.AddHeader()
```

5. Add a header with multiple rows to the table group. Takes number of sub rows.

```
IUDDTableRow tgroup1.AddHeader(int rows)
```

6. Add a row of content to the table group

```
IUDDTableRow tgroup1.AddContent()
```

7. Add content with multiple rows to the table group. Takes number of sub rows.

```
IUDDTableRow tgroup1.AddContent(int rows)
```

- IUDDTableRow interface

  Calling AddHeader() & AddContent() on the IUDDTableGroup interface provides you with the this interface. On this interface you can call the following methods

1. Set an ID for the table row for internal links.

```
trow1.SetID("id")
```

2. Set alignment information . can be "center", "left" and "right".

```
trow1.SetAlignment("center")
```

3. Set cell text. Can be cell content or header text. Takes a column index and a text string. It is added to the first row.

```
IUDDTextElement trow1.Add(int column, string text)
```

4. Set cell text. Can be cell content or header text. Takes a column index, row index and a text string. Takes in a row number because a table row can have multiple sub rows.

```
IUDDTextElement trow1.Add(int column, int subrow, string text)
```

5. Set cell content. Takes a column index and an int value. It is added to the first row.

```
IUDDTextElement trow1.Add(int column, int value)
```

6. Set cell content. Takes a column index, row index and a int value.

```
IUDDTextElement trow1.Add(int column, int subrow, string text)
```

7. Set cell text. Takes a column index and a double value. It is added to the first row.
   ```
   IUDDTextElement trow1.Add(int column, double value)
   ```

8. Set cell text. Takes a column index, row index and a double value.
   ```
   IUDDTextElement trow1.Add(int column, int subrow, double value)
   ```

9. Set cell text spanning 2 columns. Can be cell content or header text. Takes a sub row index , starting column index., ending column index and a text string.
   ```
   IUDDTextElement trow1.AddSpanningcolumnst(int subrow, int
   columnstart, int columnend, string text)
   ```

10. Set cell text. Can be cell content or header text. Takes a column index, starting sub row index, ending sub row index and a text string.
    ```
    IUDDTextElement trow1.AddpanningRows(int column, int subrowstart,
    int subrowend, string text)
    ```

- IUDDTableRow interface

  Calling Add() on the IUDDTableGroup interface provides you with the this interface. On this interface you can call the following methods

1. Set an ID for the table row for internal links.
   ```
   trow1.SetID("id")
   ```

2. Set alignment information . can be "center", "left" and "right".
   ```
   trow1.SetAlignment("center")
   ```

Includes all the methods exposed by the IUDDText interface.

Chip Model Analyzer (CMA)

Ansys Chip Model Analyzer (CMA) is an advanced Chip Power Model (CPM) creation tool for performance checking and failure diagnosis of the Power Delivery Network (PDN). CMA can use either HSpice or Nexxim simulators. HSpice and external Nexxim require licenses, but the Nexxim Internal simulator does not.

CMA does not currently interact with Electronics Desktop. However, you can launch CMA from within Electronics Desktop by clicking **Tools > Chip Model Analyzer (CMA)**.

From there, you can import CPM files for analysis or create a pseudo CPM.

For more information about using CMA, consult the CMA training documentation by launching CMA and selecting **Help > Content**.

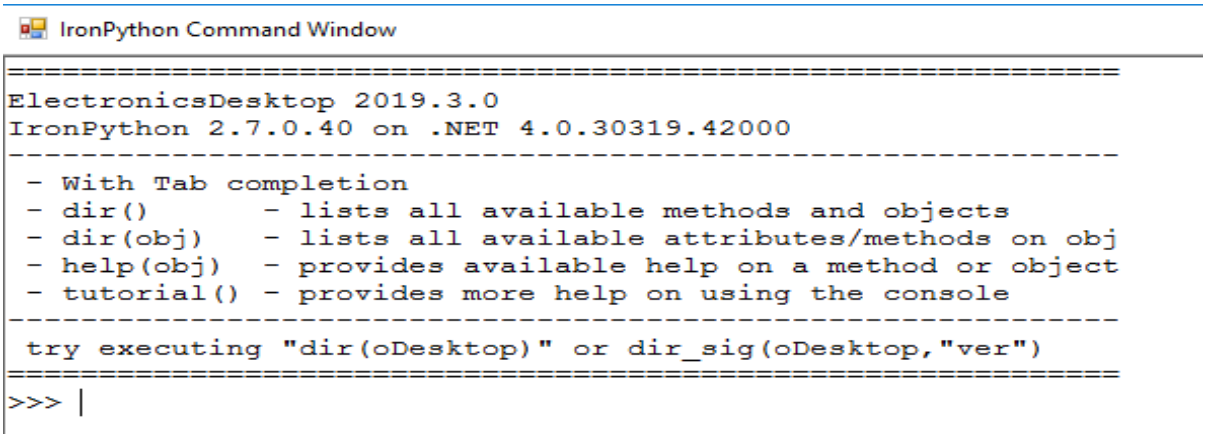
PinToPin Utility

The PinToPin utility can quickly extract HFSS or SIwave models for specified nets of package and PCB geometries. It establishes a repeatable process for robust geometry extraction, port configuration, and passive model assignment. Process execution can be done via the user interface (Windows only) or in non-graphical mode (Windows or Linux).

For more information, consult the SIwave and HFSS 3D Layout help.

24 - Desktop Scripting with IronPython

This document describes IronPython briefly and then goes on to describe the desktop provided IronPython scripting console and scripting with IronPython. You can open an IronPython Command Window by clicking **Tools>Open Command Window**.



The document assumes that you know how desktop scripting works using VBScript or Javascript.

Introduction to IronPython

IronPython Mini-cookbook

Translating Script commands from VBScript to IronPython

Scripting Using Iron Python: Putting it all Together

Standalone IronPython and Desktop IronPython

IronPython Samples

Creating User Defined Primitives and User Defined Models in Python Scripts

Related Topics

User Defined Outputs: An Introduction

Introduction to IronPython

IronPython is an implementation of the Python programming language targeting the .NET runtime. What this means in practical terms is that IronPython uses the Python programming language syntax and standard python libraries and can additionally use .NET classes and objects to give one the best of both worlds. This usage of .NET classes is fairly seamless in that a class defined in a .NET assembly can be used as a base class of a python class.

Scope

Functioning as a tutorial on Python or IronPython is way out of the scope of this document. There are several excellent resources online that do a very good job in that regard. This document only attempts to provide a limited introduction to IronPython as used to script Ansys Electromagnetics products.

This document is also not a tutorial on the scripting of Ansys Electromagnetics products. It complements the existing scripting guide (available from a product's Help menu) and provides a pythonic interpretation of that information. The reader might have to refer to either the scripting guide or recorded samples of VBScript to follow some of the sections.

Python compatibility

The version of IronPython in use is **2.6.1** and built on the .NET framework version 2.0: this version targets **Python 2.6** language compatibility. While most python files will execute under IronPython with no changes, python libraries that make use of extensions written in the C programming language (NumPy or SciPy for instance), are not expected to work under IronPython. In such cases, it might be possible to locate .NET implementation of such libraries or explore the use of IronClad.

(http://code.google.com/p/ironclad/).

Advantages of IronPython

The advantages that IronPython use provides are significant:

- Python has a large eco-system with plenty of supporting libraries, Visual IDEs and debuggers. It is actively developed and enhanced.
- IronPython, in addition, has access to the entire .NET eco system. This allows us, for instance, to create a modern GUI using the **System.Windows.Forms** assembly from IronPython code and call any other .NET assembly for that matter.
- The use of IronPython's technologies enables the ability to interactively script Desktop (feature in development). This allows better discovery of the scripting APIs as well as directly programming to the scripting API in python, a language more tractable and platform independent compared with VBScript.
- The Python syntax of dictionaries is somewhat easier to read and write when supplying arguments to the scripting methods.

Related Topics

IronPython Mini-cookbook

Translating Script commands from VBScript to IronPython

Scripting Using Iron Python: Putting it all Together

IronPython Samples

IronPython Mini-cookbook

While a tutorial on Python syntax is beyond the scope of this document, it will present simple counterparts the VBScript constructs that users tend to regularly use.

Comments

VBScript	IronPython
' Comments start with a single quote ' like this line	# Comments start with a sharp or hash # symbol, like these lines

Assigning/Creating Variables

VBScript	IronPython

' Declare with a Dim
Dim oDesktop
 	
'Assignment needs a Set instruction
Set oDesktop = oApp.GetAppDesktop ()
	# No Set syntax. Simply create and assign
	oDesktop = oApp.GetAppDesktop()

Create Lists/Arrays

VBScript	IronPython

' Declare as array of String with 11
' indices from 0 through 10
Dim myArray(0 to 10) as String
myArray(0) = "Hello"
myArray(1) = "bye"
 	
' **Declare n array with no size**
Dim array2() as String
 	
' **Re-Dimension the array once size is**
' **known**
ReDim array2(0 to 2) as String
array2(0) = "this"
array2(1) = "also"
	# Declare an empty array
	myEmptyArray = []
	
	# declare an array and initialize it with 5 ints
	myInitedArray = [1, 2, 3, 4, 5]
	
	# Python lists can have items of any type
	# and there is no pre-declaration
	# declare an array and init with mixed types
	mixed = ["hello", 1 ,2 ["nested"]]
	
	# append to an array
	mixed.append(3.5)

Create Dictionaries/Maps

VBScript	IronPython

' No direct equivalent is available as
' far as the author knows
	# an IronPython dictionary is a collection of
	# name value pairs. Just like arrays, there is
	# no restriction on the keys or the values.
	# For purposes of Ansys EM scripting however,
	# all keys must be strings
	
	# delimiters are curly braces
	# use a ":" between the key and the value
	# separate key value pairs with a ","
	myDict = {
	"a" : 1,
	"b" : "hello there",
	"c" : [1, 2, "abc"]
	}

Boolean Values

VBScript	IronPython
' Boolean literals are **in lower case** true false	# The first letter is capitalized True False

Converting Numbers to Strings and Vice Versa

VBScript	IronPython

' Use **CInt, CDbl, CBool, CLng**
' to convert the string representation
' to **the number representation. Use**
' **IsNumber to check before conversion**
Dim nStr = "100"
Dim n = CInt(nStr)
 	
' **Use CStr to convert a number to**
' its string representation
Dim v, vStr
v = 100
vStr = CStr(v)
	# use the integer() or float() or double()
	# functions to cast a string CONTAINING the
	# string representation of whatever you are
	# casting to.
	strInt = "3"
	intVal = int(strVal)
	floatVal = float(strVal)
	
	# invoke the str() function with the int/float
	# values as needed. You can alternately use
	# the string formatting method listed below
	strVal = str(42)
	strVal = str(42.345)

String formatting/concatenation

VBScript	IronPython

' string concatenation uses the &
' operator
Dim allStr, str1
str1 = " how are you"
allStr = "Hello " & " There" & str1
 	
' there seems to be no direct string
' formatting function in VBScript
' using string concatenation or using
' Replace are the two **builtin**
	# if you have two strings, you can always
	# concatenate then using the '+' operator
	str1 = "hello"
	str2 = "world"
	str12 = str1 + " " + str2
	
	# if you have different types though, string
	# and int say, you must use the string
	# formatting commands. When formatting
	# multiple arguments, they must be entered

options
Dim fmt = "{1} climbs stalk {2}"
Dim str = Replace(fmt, "{1}", "jack")
str = Replace(str, "{2"}, 10)
	# as a tuple (item1, item2,)
	num = 10
	str3 = "%s climbs stalk %d" % ("jack", num)
	str4 = "%d stalks" % num

Looping over lists

VBScript	IronPython

Dim myArray(0 to 2) as String
myArray(0) = "alpha"
myArray(1) = "bravo"
myArray(2) = "charlie"
 	
For Each i in myArray
Print i
Next
	vals = [1, 3, 3.456]
	
	def process(val):
	  return 2*val
	
	# is of the form
	# for variable_name in array
	':'
	# < indent> statement1
	# <indent> statement2
	for i in vals:
	  print i
	  print " -> " process(i)

Looping over a range

VBScript	IronPython
' Loop over a range, specify start, end ' and step For i = 0 To 10 Step 1 Print i Next	# prints out values from 0 through 9 for i in range(0, 10): print i

Related Topics

A Note About Indentation
Obtaining More Information
Discovering Methods
Help on a Method
Introduction to IronPython
Translating Script commands from VBScript to IronPython

Scripting Using Iron Python: Putting it all Together
IronPython Samples

A note about indentation

Python is one of the languages where whitespace (spaces, tabs etc) are syntactically significant. You must understand the basics of indentation before scripting in python.

Any statement that introduces a block of code should be written such that every line of the block has the same indent (leading spaces or tabs) and the indent should be at least one more than the indent of the introducing statement.

```
    # define a function that starts at 0 indentation.
    def multInt(a,b):
# every line following the def multInt which is expected
# to be a part of the function, must have the indent used
# by the first line of the function (3 spaces)
# here we introduce one more block, the if condition
# each line that belongs to the body of this func should
# have an indent that is more than the indent used by the
# if statement
If a%2 == 0:
  # I am using an indent 2 more than the parent. i.e. 5
  # spaces in total
  return (a * b) + 100
else:
  return (a * b) + 1000
```

Sample Script 1: Describing python indentation

Related Topics

Obtaining More Information
Discovering Methods
Help on a Method
Introduction to IronPython
Translating Script commands from VBScript to IronPython
Scripting Using Iron Python: Putting it all Together
IronPython Samples

Obtaining more Information on Python

Reading a book and searching online are two of the standard options. There are several very good python tutorials online and the command window is also a great way to quickly execute code and learn by doing.

Much of the official python documentation is embedded into each python distribution and the command window can also be used to get more help. You can open the IronPython Command Window by clicking **Tools>Open Command Window**. You can also aces the command window executable, **Ipy.exe**, from the **IronPython** directory under the Ansys Electromagnetics product installation directory. You can use this window to interactively enter python commands (any of the pure python samples in this document)

Related Topics

Discovering Methods

Help on a Method

Introduction to IronPython

Translating Script commands from VBScript to IronPython

Scripting Using Iron Python: Putting it all Together

IronPython Samples

Discovering Methods

For instance, if you want to list all methods available in the string module, you would type in "import string" to import the module first and then type in "dir(string)". You will get a list of all the methods available (as well as some __somename__ internal names that can be ignored)

```
>>> import string
>>> dir(string)
['Formatter', 'Template', '_TemplateMetaclass', '__builtins__', '__doc__', '__file__', '__name__',
'__package__', '_float', '_idmap', '_idmapL', '_int', '_long', '_multimap', '_re', 'ascii_letters',
'ascii_lowercase', 'ascii_uppercase', 'atof', 'atof_error', 'atoi', 'atoi_error', 'atol',
'atol_error', 'capitalize', 'capwords', 'center', 'count', 'digits', 'expandtabs', 'find',
'hexdigits', 'index', 'index_error', 'join', 'joinfields', 'letters', 'ljust', 'lower', 'lowercase',
'lstrip', 'maketrans', 'octdigits', 'printable', 'punctuation', 'replace', 'rfind', 'rindex', 'rjust',
'rsplit', 'rstrip', 'split', 'splitfields', 'strip', 'swapcase', 'translate', 'upper', 'uppercase',
'whitespace', 'zfill']
```

Related Topics

Obtaining More Information

Help on a Method

Introduction to IronPython

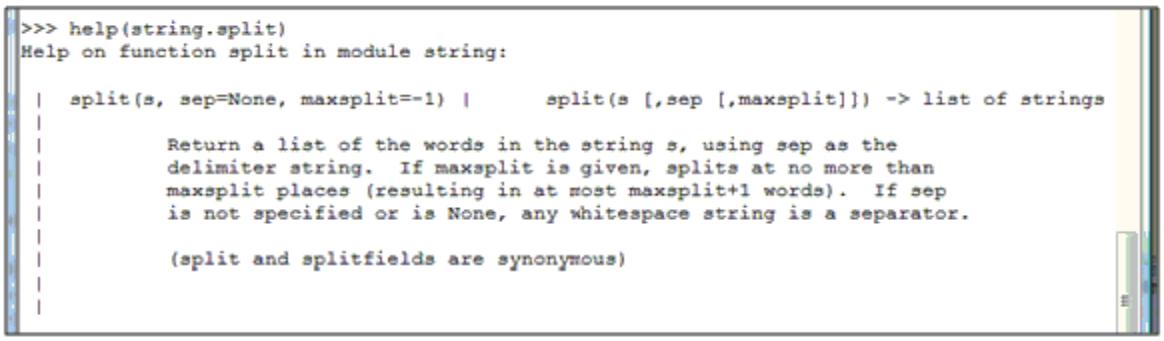
Translating Script commands from VBScript to IronPython

Scripting Using Iron Python: Putting it all Together

IronPython Samples

Help on a Method

Once you know a function name (maybe you see something you like from the dir(string) listing above), you can get more help on it using the builtin **help** method. For instance, executing **help (string.split)** in the **IronPython Command Window** displays the following:



Related Topics

Obtaining More Information

Discovering Methods

Introduction to IronPython

Translating Script commands from VBScript to IronPython

Scripting Using Iron Python: Putting it all Together

IronPython Samples

Translating Script commands from VBScript to IronPython

This chapter briefly describes Desktop scripting methods and arguments via VBScript samples. The distinctions made here are significant and come in use later when translating scripts written in VBScript to IronPython.

Script Method Argument

VBscript Method Call Types

VBScript Sub-Routines

Converting VBScript Function calls to IronPython Syntax

Related Topics

Introduction to IronPython
IronPython Mini-cookbook
Scripting Using Iron Python: Putting it all Together
IronPython Samples

Script Method Argument

Script method calls in VBscript are described later in this document, they generally take the form:

```
objectName .methodName ( arg1, arg2, ..)
```

The function call syntax is a standard followed by several programming languages, however, the argument types in VBScript objects, when used for product scripting, are restricted to the following

Primitive types
Named Array
Named Function

Primitive Types

Primitive types are the standard **bool, int, float, double** and **string**

Named Array

Named arrays are a special construct used very commonly and can be found in practically every recorded script sample.

It starts with **Array("NAME:someName**" and is followed by a collection of comma separated values which can be:

- A primitive value
- An array of primitive values
- Another named array
- A key, of the form "**keyName:="** followed by
  - A primitive value
  - A function (described next)

Named Function

Named functions are arrays which start with **Array(** and do not have a leading "NAME:name" item. **They are always introduced by a key** and can contain comma separated values of the following type:

- A primitive value
- A key (of the form "**keyName:="**) followed by

- A primitive value
- Another function (nested function)

Related Topics

Translating Script commands from VBScript to IronPython

VBscript Method Call Types

VBScript method calls fall into two categories and the distinction between the two results in syntax differences. These syntax differences are significant when converting VBScript to IronPython.

VBScript Functions

In VBScript terminology functions return values. The syntax for this is the one shared with practically all programming languages.

```
Set oDesktop = oAnsoftApp.GetAppDesktop()
Set oProject = oDesktop.NewProject
```

Sample Script 2: VBScript function call sample

Note that the significant item here is that the method name is **always** followed by an argument list enclosed in parentheses **if there are arguments**. If the argument list is empty as shown above for the *NewProject* call, the parentheses can be omitted.

VBScript Sub-Routines

VBScript Sub-Routines are those that do not have any return value. VBScript allows these to be written without any parentheses even if they have a non-empty argument list.

```
oModule.CreateReport "XY Plot1", "Standard", "XY Plot", "optimtee :
optimtee", _
    Array("Domain:=", "Sweep"), Array("Freq:=", Array("All"),
    "offset:=",_
    Array("0uin")), Array("X Component:=", "Freq", "Y Component:=", _
    Array("dB20(S(1,1))", "dB20(S(1,2))", "dB20(S(1,3))", _
     "dB20(S(2,1))", "dB20(S(2,2))", "dB20(S(2,3))", "dB20(S(3,1))",
     "dB20(S(3,2))", "dB20(S(3,3))")), Array()
```

Sample Script 3: VBScript Sub-Routine sample

Related Topics

Translating Script commands from VBScript to IronPython

Converting VBScript Function calls to IronPython Syntax

When converting functions, the important point to remember is that IronPython function names, when used for scripting, are **always** followed by parentheses. So:

- If you see a VBScript snippet that looks like a VBScript Sub-Routine, remember to add parentheses.
- If you see a VBScript function that has no arguments and no parenthesis, remember to add them around an empty argument list.

The parentheses change is the only one to keep in mind when converting VBScript function calls syntax to IronPython.

Return Values

Primitive Method Arguments

Named Array Argument

Named Array Values with All Key Value Pairs

Named Arrays with Nested Named Arrays

Function Blocks

Return Values

VBscript return values are sometimes assigned via the **Set** declaration. IronPython return values are simple assignment (see the cookbook chapter)

Related Topics

Converting VBScript Function calls to IronPython Syntax

Primitive Method Arguments

Replace each VBScript primitive with an equivalent IronPython primitive. The main thing to notice here is that Boolean values in IronPython have their first letter capitalized.

```
True instead of true and False instead of false
```

Related Topics

Converting VBScript Function calls to IronPython Syntax

Named Array Argument

The recommended approach here is to simply replace a VBScript array with a python array. The mapping is quite simple:

- Change **Array(** to **[** and close with a **]** instead of the **)**
- Remove the line continuation symbols: **_**

- Map Boolean values correctly

```
oEditor.CreateCone Array("NAME:ConeParameters", "XCenter:=", "0mm", _
"YCenter:=", "0mm", "ZCenter:=", "0mm", "WhichAxis:=", "Z", "Height:=", "2mm",_
"BottomRadius:=", "1.56204993518133mm", "TopRadius:=", "0mm"), Array("NAME:Attributes",
"Name:=", "Cone1", "Flags:=", "", "Color:=", _
"(132 132 193)", "Transparency:=", 0, "PartCoordinateSystem:=", _
"Global", "UDMId:=", "", "MaterialValue:=", _
"" & Chr(34) & "vacuum" & Chr(34) & "", "SolveInside:=", true)
```

Sample Script 4: Create cone recorded snippet

For instance, method and named VBScript arrays in the snippet above are translated to

```
    oEditor.CreateCone(
  [
    "NAME: ConeParameters",
    "XCenter:=" , "0mm",
    "YCenter:=" , "0mm",
    "ZCenter:=" , "0mm",
    "WhichAxis:=" , "Z",
    "Height:=" , "2mm",
    "BottomRadius:=" , "1.56204993518133mm",
    "TopRadius" , "0mm"
  ],
  [
    "NAME:Attributes",
    "Name:=" , "Cone1",
    "Flags:=" , "",
    "Color:=" , "(132 132 193)",
    "Transparency:=" , 0,
    "PartCoordinateSystem:=" , "Global",
    "UDMId:=" , "",
    "MaterialValue:=" , "\"vacuum\"",
    "SolveInside:=" , True
    ])
```

Sample Script 5: Create a cone in IronPython

Note that the formatting (which helps readability immensely) is not really needed. All that **had** to be done was

- Add the parentheses since the VBScript subroutine omits it
- Replace the **Array( )** delimiters with **[ ].**
- Remove the **Char(34)** function (which introduced a double quote) and replace it with the escaped double quote **\"** literal.
- Replace **true** with **True**
- Remove the line continuation symbol, **_**

Related Topics

Converting VBScript Function calls to IronPython Syntax

Named Array Values with All Key Value Pairs

While it is in general not allowed to replace the arrays and nested arrays with python dictionaries, in the case where the named array consists entirely of key value pairs (like the sample above), one can use a dictionary and **avoid typing the trailing ":=" symbols after the keys**. This further aids readability of the script.

- If all key value pairs
- Remove the trailing ":=" after each key
- Replace the "," after the key with a ":"
- If the named array is the top level argument, ensure that the "NAME:name" is present and is split into "NAME" : "name" as a key value pair
- Enclose the converted array in a { } pair to declare the dictionary.

```
   oEditor.CreateCone(
{
"NAME" : "ConeParameters",
"XCenter" : "0mm",
"YCenter" : "0mm",
"ZCenter" : "0mm",
"WhichAxis" : "Z",
"Height" : "2mm",
"BottomRadius": "1.56204993518133mm",
"TopRadius" : "0mm"
},
{
"NAME" : "Attributes",
"Name" : "Cone1",
"Flags" : "",
"Color" : "(132 132 193)",
"Transparency" : 0,
```

```
"PartCoordinateSystem": "Global",
"UDMId" : "",
"MaterialValue" : "\"vacuum\"",
"SolveInside" : True
}
```

Sample Script 6: CreateCone in IronPython using dictionary syntax

Related Topics

Converting VBScript Function calls to IronPython Syntax

Named Arrays with Nested Named Arrays

- Split the "NAME:name" field into a key value pair
- Translate array key value pair to a dictionary key value pair.
- Create a new key with the name of the nested array and keep the nested array (as an array or as a dictionary) as its value. If the nested array is being retained **as an array**, **the**"NAME**:name**" **field should be retained in the array**. If the nested array is being converted to a dictionary, the name is optional: if also retained in the nested array, it must match the outer key.

```
   [ "NAME:name",
"key1:=" , 1,
"key2:=" , 2,
["NAME:name2", "R:=", 255]
   ]
```

Sample Script 7: Named array with nested named array in array syntax

The above named array with a nested named array (after conversion to IronPython as named array) can be converted to a dictionary as well. The dictionary can take any of the following forms

```
   { "NAME" : "name",
"key1" : 1,
"key2" : 2,
"name2" : ["NAME:name2", "R:=", 255]
   }
```

Sample Script 8: Named array with nested named array as mixed dictionary + array

```
   { "NAME" : "name",
   "key1" : 1,
   "key2" : 2,
```

```
    "name2" : {"R" : 255}
    }
```

Sample Script 9: Named array with nested named array in all dictionary syntax

```
    { "NAME" : "name",
    "key1" : 1,
    "key2" : 2,
    "name2" : {
   "NAME" : "name2",
   "R" : 255
}
    }
```

Sample Script 10: Nested named array with optional "NAME:" field

Related Topics

Converting VBScript Function calls to IronPython Syntax

Function Blocks

Function blocks in VBScript argument syntax are represented as arrays without the "NAME:.." field. However, functions are always introduced by a key in a parent structure. Function blocks can therefore never exist as a top-level argument. They are only found as the value pairs inside a named array or inside another function block.

Note	Function blocks and their items cannot be converted to dictionaries even though they might be composed entirely of key value pairs.

The reason for this is the need to main the user-entered order. Every item in a function block is expect to be transmitted to the script method in exactly the same order as typed out and this is impossible to achieve when a dictionary is used (as the keys get reordered according to the dictionary's internal tree/key sorting scheme).

Note	When you see a function block, simply replace the Array( ) delimiters with the python array delimiters [ ]

Related Topics

Converting VBScript Function calls to IronPython Syntax

Scripting Using Iron Python: Putting it all Together

If you have existing VBScript/Javascript scripts use existing scripts them as much as possible by either embedding the test into the IronPython script or invoking them.

Translating a script in VBScript to IronPython

Read the chapter on translation and study the samples in that chapter as well as those in the appendix. For python syntax and the differences, the mini-cookbook chapter will also be useful.

Writing an IronPython script from scratch

Read through the scripting guide available from the product's help menu and translate the VBScript methods described to IronPython using the information provided in the chapter on translation. Studying the samples in the document will also prove helpful.

For python syntax and the differences, the mini-cookbook chapter will also be useful.

IronPython Script Execution Environment

Scripting using Embedded VBScript or JavaScript

Scripting with IronPython

Related Topics

Introduction to IronPython

IronPython Mini-cookbook

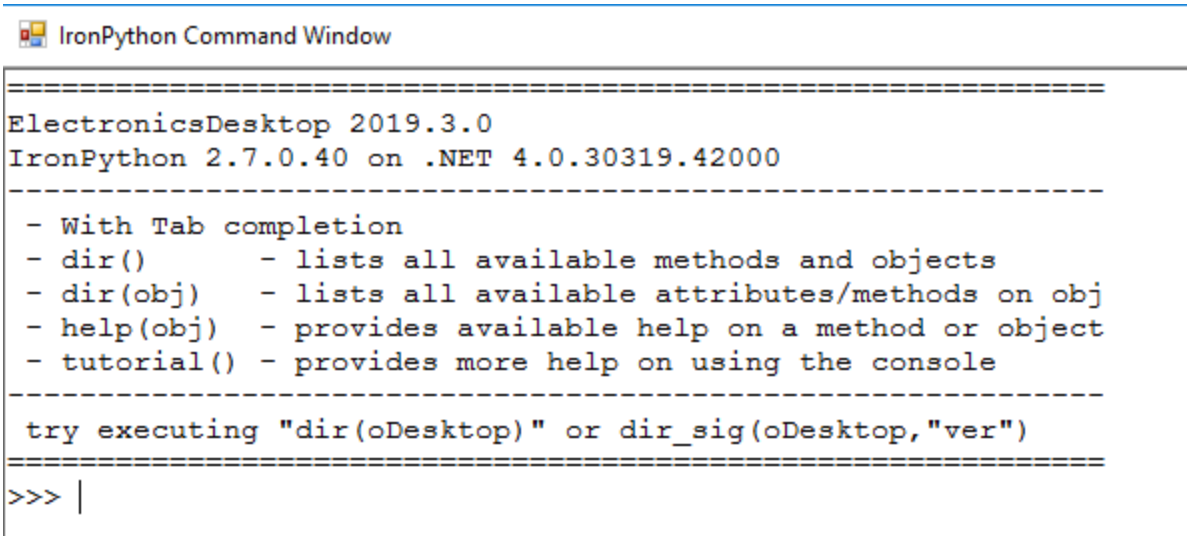
Translating Script commands from VBScript to IronPython

Appendix: IronPython Samples

IronPython Script Execution Environment

Scripts written in IronPython are executed by desktop in four different ways:

- **Tools>Open Command Window**, to open the **IronPython Command Window:**



- **Tools > Run Script** menu item, select "IronPython" from the file type drop-down list.
- Launch the product with a script argument.
  - Maxwell -runscript someScript.py to keep Maxwell GUI open after completing script execution.
  - Maxwell -features=beta -ng -runscriptandexit someScript.py to run Maxwell in a non-graphical mode and exit after script completion. Note that this is a beta feature.
- Register an IronPython script as an external tool using the **Tools > External Tools** menu item.

When desktop executes a script, it does so in an execution environment setup with predefined variables and functions. These predefined variables and functions are how the script communicates with the desktop and they come in three flavors.

Script Argument for IronPython

Script Argument for IronPython

When scripts are launched using the **Tools > Run Script** menu item, the dialog that pops up allows the user to specify arguments.

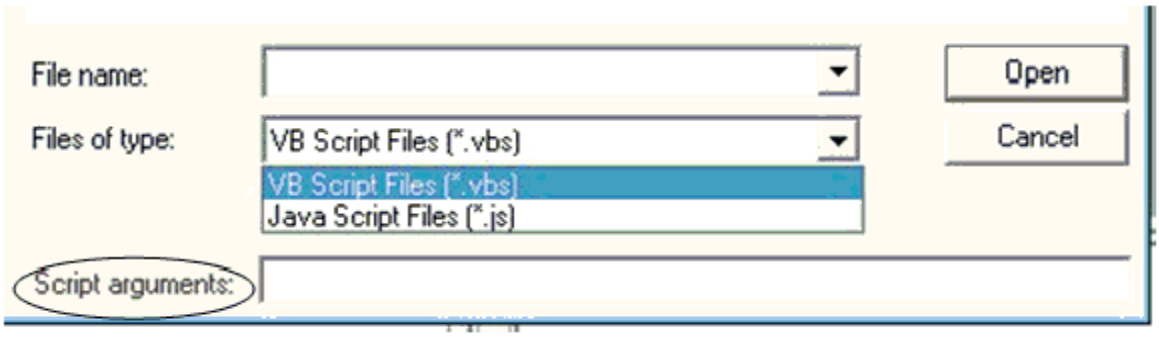


Figure 1: Run Script dialog and script arguments

Any argument specified here is communicated to the script being executed as the predefined variable **ScriptArgument**.

Related Topics

IronPython Script Execution Environment

Scripting using Embedded VBScript or JavaScript

Since script recording is still done in VBScript and users are expected to have a significant collection of VBScript or JavaScript assets, it is useful to continue to use existing script files and snippets even when scripting in IronPython. The various **Run<*>Command** methods have been designed for this purpose.

For instance: one can create a parameterized cone in HFSS by executing the following IronPython script from the **Tools>Run Script** menu.

```
# assign the VBScript snippet obtained from a script recording from
HFSS to
# coneScript and replace the BottomRadius recorded value with botRadius
coneScript = """Dim oAnsoftApp
Dim oDesktop
Dim oProject
Dim oDesign
Dim oEditor
Dim oModule
Set oAnsoftApp = CreateObject("AnsoftHfss.HfssScriptInterface")
Set oDesktop = oAnsoftApp.GetAppDesktop()
oDesktop.RestoreWindow
Set oProject = oDesktop.GetActiveProject()
oProject.InsertDesign "HFSS", "HFSSPyTestDesign", "DrivenModal", ""
Set oDesign = oProject.SetActiveDesign("HFSSPyTestDesign")
Set oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.CreateCone Array("NAME:ConeParameters", _
 "XCenter:=", "0mm", "YCenter:=", "0mm", "ZCenter:=", "0mm",_
 "WhichAxis:=", "Z", "Height:=", "2mm", _
 "BottomRadius:=", "3mm", _
 "TopRadius:=", "0mm"), Array("NAME:Attributes", "Name:=", _
 "Cone1", "Flags:=", "", "Color:=", "(132 132 193)", "Transparency:=",
0, _
```

```
 "PartCoordinateSystem:=", "Global", "UDMId:=", "", "MaterialValue:=",
_
 "" & Chr(34) & "vacuum" & Chr(34) & "", "SolveInside:=", _
 true)
"""
SetScriptingLanguageToVBScript()
RunScriptCommand(coneScript)
```

Sample Script 11: Hybrid VBScript + IronPython scripting: parameterized Cone Creation

Even though recorded VBScript is used for scripting, the incremental functionality that is provided using IronPython is the ability to write a GUI using IronPython/.NET, collect information from the user and then modify or generate the VBScript commands to actually script the Ansys Electromagnetics desktop. This GUI functionality is cross platform and a significant positive. The following example demonstrates a contrived use of a .NET window form to display the argument supplied to the IronPython script (via the ***ScriptArgument*** variable).

```
        #import the CLR references
        import clr
        clr.AddReference("System.Windows.Forms")

        from System.Windows.Forms import Application, Form, Label,
        Button, DockStyle

        # the GUI form to show some text
        # the class below derives from From (System.Windows.Forms.Form)
        # imported above from the .NET assembly.
        class ShowPropertiesForm(Form):
    def __init__(self, name, text):
      self.Name = name
      self._label = Label()
      self._label.Text = text
      self._label.Dock = DockStyle.Fill

      _button = Button()
      _button.Text = "Close"
      _button.Dock = DockStyle.Bottom
      _button.Click += self._buttonPressed

      self.Controls.Add(self._label)
```

```
    self.Controls.Add(_button)

def _buttonPressed(self,sender,args):
    self.Close()

    #----------------------------------------------------
    # Main script code
    #----------------------------------------------------
    #display the ScriptArgument variable as the text label
    # in the form.
    gui = ShowPropertiesForm("Sample Form", ScriptArgument)

    # This makes it a modal dialog.
    gui.ShowDialog()

    # the following will make it a non-modal dialog
    #Application.Run(gui)
```

Sample Script 12: Demonstrates the use of a .NET form from IronPython

While creating cross platform user interfaces from scripts is one of the main motivations driving the adoption of IronPython, any .NET assembly can be used with the caveat that Linux use requires Mono compatibility of any used assemblies.

While this hybrid approach is useful when you have existing VBScript commands that you want to reuse or when you want to quickly parameterize a recorded sample, the one significant limitation of this approach is the inability to capture return values from VBScript or JavaScript calls that do return something. Full two way communication with the product requires the use of pure IronPython to directly invoke the script objects as described below.

Related Topics

IronPython Script Execution Environment

Scripting with IronPython

While this section talks about directly interacting with the script objects, note that you can execute VBScript or Javascript at any point using any of the available Run*Command functions. using your existing script assets in this fashion and mixing with IronPython code for new functionality as needed is a viable and option.

Access to the application scripting objects is provided via the predefined **oDesktop** object (as listed in Script Objects). Interacting with the script objects is very natural, method calls are made just like in VBScript except that the argument syntax is somewhat simplified to follow natural

Python syntax. All primitive types (string, integer, double) map to the natural primitive types in python. The only differences from the VBScript syntax are seen when specifying array type arguments. The differences are described in earlier chapters.

Note	The typical VBScript calls to obtain the registered COM scripting interface via CreateObject calls and then obtain the oDesktop object from it using the GetAppDesktop() is not needed (or even supported on all platforms). Since all scripting occurs in the context of a running workbench, the available Desktop object is always provided and expected to be used directly.

Scripting using the IronPython scripting API is very much like scripting with VBScript except that

- Any argument is supplied via the built in **ScriptArgument** variable
- The **oDesktop** object is always available
- The scripting method names are identical to the ones used with VBScript
- Method calls, while the name is the same have to adhere to the rule of ensuring trailing parentheses irrespective of whether the function returns anything or has any arguments.
- Any compound/block arguments should be translated to the appropriate IronPython array or dictionary syntax.

The samples section lists a collection of pure IronPython snippets: these, along with the various script snippets listed in this document should serve as a guide and reference.

Related Topics

IronPython Script Execution Environment

Standalone IronPython and Desktop IronPython

In general, it is easier to run a script directly from Electronics Desktop. Standalone IronPython does not implement all the functionality available when a script is run from Electronics Desktop. It only implements full support for COM functions.

Running Standalone IronPython

Standalone IronPython uses COM to get the handle to the AnsysEDT app. To run standalone IronPython, you'll need to call the IronPython interpreter ipy64.exe.

It is located in:

```
<AnsysEDTInstallationPath>\<version>\<platform>\common\IronPython\ipy64.exe
```

For example, to run myScript.py, type the following in the command line:

```
"C:\Program Files\AnsysEM\v231\Win64\common\IronPython\ipy64.exe" "<filePath>\myScript.py"
```

You can set the interpreter to be the default program when double-clicking the .py script. You can use any recorded script as the basis for a standalone script and simply add an installation-internal path to the python module search path (as shown below) and end the script with a new shutdown call.

Using a Recorded Script

A python script recorded in AnsysEDT already has the required lines to be run as a standalone, except for the first two lines (path settings) and the final `Shutdown()` call. See the example script below.

Creating an External Script

When creating a script outside of Electronics Desktop, the following lines should be included at the beginning of your script:

- `import sys`

  # Imports the sys module containing system-specific functions native to IronPython.
- `sys.path.append("<InstallationPath>")`

  # Adds the Electronics Desktop installation path to the list of directories Python searches for modules and files.
- `sys.path.append("<InstallationPath>/PythonFiles/DesktopPlugin")`

  # Adds the PythonFiles/DesktopPlugin subfolder to the list of directories Python searches for modules and files.
- `import ScriptEnv`

  # This imports ScriptEnv.py from the installation path specified above. ScriptEnv.py performs an operating system check and defines functions used in Electronics Desktop scripts. See the annotations in the ScriptEnv.py file for more information.
- `ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")`

  or `ScriptEnv.InitializeNew(NonGraphical=True)`

  # `Initialize` and `InitializeNew` are functions within ScriptEnv.py. The first option launches Electronics Desktop. The second allows you to run a script without launching Electronics Desktop. See the annotations in the ScriptEnv.py file for more information.

You must end the script with:

- `ScriptEnv.Shutdown()`

  # This stops ScriptEnv.py. If you are running multiple scripts, include this only at the end of the last script.

Example Script

```
import sys
sys.path.append(r"C:\Program Files\AnsysEM\v231\Win64")
sys.path.append(r"C:\Program
Files\AnsysEM\v231\Win64\PythonFiles\DesktopPlugin")

import ScriptEnv
ScriptEnv.Initialize("Ansoft.ElectronicsDesktop")
```

```
oDesktop.RestoreWindow()
oProject = oDesktop.NewProject()
oProject.InsertDesign("HFSS", "HFSSDesign1", "DrivenModal", "")
oDesign = oProject.SetActiveDesign("HFSSDesign1")
oEditor = oDesign.SetActiveEditor("3D Modeler")
oEditor.CreateRectangle(
[
"NAME:RectangleParameters",
"IsCovered:= ", True,
"XStart:= ", "-0.2mm",
"YStart:= ", "-3mm",
"ZStart:= ", "0mm",
"Width:= ", "0.8mm",
"Height:= ", "1.2mm",
"WhichAxis:= ", "Z"
],
[
"NAME:Attributes",
"Name:= ", "Rectangle1",
"Flags:= ", "",
"Color:= ", "(132 132 193)",
"Transparency:= ", 0,
"PartCoordinateSystem:=", "Global",
"UDMId:= ", "",
"MaterialValue:= ", "\"vacuum\"",
"SolveInside:= ", True
])
oDesign.SetDesignSettings(['NAME:Design Settings Data', 'Allow Material
Override:=', True, 'Calculate Lossy Dielectrics:=', True])
oEditor.SetModelUnits(['NAME:Units Parameter', 'Units:=', 'mil',
'Rescale:=' , False ])
ScriptEnv.Shutdown()
```

IronPython Samples

Change property

The following snippets show how a change property command (in this case, to change the color of a cone) looks in VBScript and its two possible IronPython variants.

```
   oEditor.ChangeProperty Array("NAME:AllTabs", Array
   ("NAME:Geometry3DAttributeTab",_
Array("NAME:PropServers", "Cone1"), _
Array("NAME:ChangedProps", _
Array("NAME:Color", "R:=", 255, "G:=", 255, "B:=", 0))))
```

Sample Script 13: ChangeProperty command to change color of a cone in VBScript

```
   oEditor.ChangeProperty(
   ["NAME:AllTabs",
["NAME:Geometry3DAttributeTab",
["NAME:PropServers", "Cone1"],
["NAME:ChangedProps",
["NAME:Color", "R:=", 0, "G:=", 0, "B:=", 64]
]
   ]
   ])
```

Sample Script 14: ChangeProperty command to change color of cone using Python arrays

Any time there are named arrays composed purely of key-value pairs, they can always be represented using a Python dictionary, irrespective of the nesting of said named array.

```
   oEditor.ChangeProperty(
["NAME:AllTabs",
["NAME:Geometry3DAttributeTab",
  ["NAME:PropServers", "Cone1"],
  ["NAME:ChangedProps",
  {
  "NAME":"Color",
  "R" : 0,
"G" : 64,
"B" : 0
}]]
   ])
```

Sample Script 15: ChangeProperty command to change the color of a cone using Python arrays and dictionaries

Create a Cone using IronPython

Most scripting tasks using IronPython are expected to be formatted as the following example. One starts with the predefined **oDesktop** object and drills down to the design, editors, modules etc and issues any required commands on the object while formatting the script command arguments in natural python syntax.

```
    oProject = oDesktop.GetActiveProject()
    oDesign = oProject.InsertDesign
    ("HFSS","Random","DrivenModal","")
    oEditor = oDesign.SetActiveEditor("3D Modeler")
    oEditor.CreateCone(
    {
"NAME" : "ConeParameters",
"XCenter" : "0mm",
"YCenter" : "0mm",
"ZCenter" : "0mm",
"WhichAxis" : "Z",
"Height" : "2mm",
"BottomRadius" : "1.56204993518133mm",
"TopRadius" : "0mm"
},
{
"NAME" : "Attributes",
"Name" : "Cone1",
"Flags" : "",
"Color" : "(132 132 193)",
"Transparency" : 0,
"PartCoordinateSystem": "Global",
"UDMId" : "",
"MaterialValue" : "\"vacuum\"",
"SolveInside" : True
}
)
```

Sample Script 16: IronPython script to create a cone

Create geometry and then create a grid from it using copy/paste/move

The following script demonstrates slightly more advanced use of scripting and the use of return values from script methods. It creates a 5x5 grid of cones and also demonstrates the adding of information messages to the application's message window.

```
    oProject = oDesktop.GetActiveProject()
    oDesign = oProject.InsertDesign("HFSS","Hersheys
    Kisses","DrivenModal","")
    oEditor = oDesign.SetActiveEditor("3D Modeler")


    # create the first cone
    AddInfoMessage("Creating first cone")
    firstConeName = "firstCone"
    coneBotRad = "1.5mm"
    oEditor.CreateCone(
[
"NAME" : "ConeParameters",
"XCenter" : "0mm",
"YCenter" : "0mm",
"ZCenter" : "0mm",
"WhichAxis" : "Z",
"Height" : "2mm",
"BottomRadius": coneBotRad,
"TopRadius" : "0mm"
],
[
"NAME" : "Attributes",
"Name" : firstConeName,
"Flags" : "",
"Color" : "(132 132 193)",
"Transparency" : 0,
"PartCoordinateSystem": "Global",
"UDMId" : "",
"MaterialValue" : "\"vacuum\"",
"SolveInside" : True
]
)

    # Now replicate this a few times and create an array out of it
```

```
        AddInfoMessage("Replicating it 24 times")
        for x in range(5):
    for y in range(5):
# leave the first one alone in it's created
# position
        if x == 0 and y == 0:
continue

        # all other grid positions, replicate from the
        # first one

        # copy first
        oEditor.Copy(
    {
    "NAME" : "Selections",
    "Selections" : firstConeName
    }
        )

        # paste it and capture the pasted name
        # the pasted names come in an array as we could
        # be pasting a selection cmposed of multiple objects
        pasteName = oEditor.Paste()[0]

        # now move the pasted item to it's final position
        oEditor.Move(
    {
    "NAME" : "Selections",
    "Selections" : pasteName
    },
    {
    "NAME" : "TransalateParameters",
    "CoordinateSystemID" : -1,
    "TranslateVectorX" : "%d * 3 * %s" % (x, coneBotRad),
    "TranslateVectorY" : "%d * 3 * %s" % (y, coneBotRad),
    "TranslateVectorZ" : "0mm"
    }
```

```
)


# Now fit the display to the created grid
oEditor.FitAll()
```

Sample Script 17: Sample script to create a cone and then use copy/paste/move to replicate it.

Related Topics

Introduction to IronPython

IronPython Mini-cookbook

Translating Script commands from VBScript to IronPython

Scripting Using Iron Python: Putting it all Together

Creating User Defined Primitives and User Defined Models in Python Scripts

You can create User Defined Primitives and User Defined Models in Python scripts (based on the IronPython implementation).

For more information, consult the Maxwell Scripting Guide.

25 - Optimetrics

Optimetrics enables you to determine the best design variation among a model's possible variations. You create the original model, the *nominal design*, and then define the design parameters that vary, which can be nearly any design parameter assigned a numeric value in Maxwell. For example, you can parameterize the model geometry or material properties. You can then perform the following types of analyses on your nominal Maxwell design:

Parametric	In a parametric analysis, you define one or more *variable sweep definitions,* each specifying a series of variable values within a range. For example, you can parameterize component values. (See Variables in Maxwell for more information.) Optimetrics solves the design at each variation. You can then compare the results to determine how each design variation affects the performance of the design. Parametric analyses are often used as precursors to optimization solutions because they help to determine a reasonable range of variable values for the optimization analysis.
Optimization	For an optimization analysis, you identify the cost function and the optimization goal. Optimetrics changes the design parameter values to meet that goal. The cost function can be based on any solution quantity that Maxwell can compute.
Sensitivity	In a sensitivity analysis, you use Optimetrics to explore the vicinity of the design point to determine the sensitivity of the design to small changes in variables.
Tuning	Tuning allows you to change variable values interactively while monitoring the performance of the design.If you want to ensure that tuning does not resolve variations already solved by parametric setup, you must check **Save Fields Mesh** in the **Options** tab of the optimetrics setup.
Statistical	In a statistical analysis, you use Optimetrics to determine the distribution of a design's performance, which is caused by a statistical distribution of variable values.
Design of Experiments	Design of Experiments involves the creation of a response surface that depicts and predicts how variable values will affect the design.
DesignXplorer	An optimization tool for studying a range of design variations, used with Design of Experiments.
opiSLang	optiSLang simulations can be integrated with Ansys Electronics Desktop, such that you can create a setup very much like an Optimetrics setup to run through an optisLang installation.

Note	Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

Related Topics

Setting up a Parametric Analysis
Setting up an Optimization Analysis
Setting up a Sensitivity Analysis
Tuning a Variable
Setting up a Statistical Analysis
Parametric Overview
Optimization Overview
Sensitivity Analysis Overview
Statistical Analysis Overview
Tuning Overview
Using Distributed Analysis
Options: Optimetrics Options

Parametric Overview

Running a parametric analysis enables you to simulate several design variations using a single model. You define a series of variable values within a range, or a variable sweep definition, and Maxwell generates a solution for each design variation. You can then compare the results to determine how each design variation affects the performance of the design.

You can vary design parameters that are assigned a quantity, such as geometry dimensions, material properties, and boundary and excitation properties. (See the help topic for the specific parameter you want to vary.) The number of variations that can be defined in a parametric sweep setup is limited only by your computing resources.

To perform a parametric analysis, you first create a nominal design. A nominal design is created like any other design, except that variables are assigned to those aspects of the model you want to change. All variables must be defined before you start the parametric analysis. Although you are not required to solve the nominal design before performing a parametric analysis, doing so helps ensure that the model is set up and operates as intended. Alternatively, you can perform a validation check on the nominal design before performing a parametric analysis.

Parametric analyses are often used as precursors to optimization analyses because they enable you to determine a reasonable range of variable values for an optimization analysis.

Related Topics

Setting up a Parametric Analysis

Setting Up a Parametric Analysis

A *parametric setup* specifies all of the design variations that Optimetrics drives Maxwell to solve. A parametric setup is made up of one or more *variable sweep definitions*, which are a set of variable values within a range that you want Maxwell to solve when you run the parametric setup.

You can define more than one parametric setup per design.

Note	Once you have created a parametric setup, you can copy and paste it, and then make changes to the copy, rather than redoing the whole process for minor changes.

To add a parametric setup to a design:

1. On the **Maxwell3D or Maxwell2D** menu, point to **Optimetrics Analysis**, and then click

   Add Parametric .

   - Alternatively, right-click **Optimetrics** in the project tree, and then click **Add>Parametric** on the shortcut menu.

   The **Setup Sweep Analysis** dialog box appears.
2. Add a variable sweep definition.

After you define a parametric sweep, a shortcut menu becomes available when you right-click the setup name.

Note	Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

Related Topics

Adding a Variable Sweep Definition

Specifying a Solution Setup for a Parametric Setup

Using Distributed Analysis

Parametric Overview

Adding a Variable Sweep Definition

A parametric setup is made up of one or more *variable sweep definitions*. A variable sweep definition is a set of variable values within a range that Optimetrics drives Maxwell to solve when the parametric setup is analyzed. You can add one or more sweep definitions to a parametric setup.

Note	Sweeping a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

1. On the **Maxwell3D or Maxwell2D** menu, point to **Optimetrics Analysis**, and then click

   Add Parametric .

- Alternatively, right-click **Optimetrics** in the project tree, and then click **Add>Parametric** on the shortcut menu.

The **Setup Sweep Analysis** dialog box appears.

2. Under the **Sweep Definitions** tab, click **Add**.

   The **Add/Edit Sweep** dialog box appears.

   All the independent variables associated with the design are listed in the **Variable** pull-down list of the **Add/Edit Sweep** dialog.

3. Click the variable for which you are defining the sweep definition from the **Variable** pull-down list.

   If you do not define a sweep definition for a variable in the list, the variable's current value in the nominal design is used in the parametric analysis.

4. Specify the variable values to be included in the sweep.
5. Click **Add**, and then click **OK**.

   You return to the **Setup Sweep Analysis** dialog box. The variable sweep is listed in the top half of the window.

6. View the design variations that are to be solved in table format under the **Table** tab. Viewing the sweep definition in table format enables you to visualize the design variations that are to be solved and manually adjust sweep points if necessary.
7. Optionally, click the button for setting **HPC and Analysis Options**, which allows you to select or create an analysis configuration.
8. Click **OK**.

Related Topics

Specifying Variable Values for a Sweep Definitions

Synchronizing Variable Sweep Definitions

Modifying a Variable Sweep Definition Manually

Overriding a Variable's Current Value in a Parametric Setup

Specifying Variable Values for a Sweep Definition

To specify the variable values to include in a sweep definition:

1. Select one of the following in the Add/Edit Sweep dialog box:

Single value	Specify a single value for the sweep definition.
Linear step	Specify a linear range of values with a constant step size.
Linear count	Specify a linear range of values and the number, or count of points within this range.
Decade count	Specify a logarithmic (base 10) series of values, and the number of values to calculate in each decade.
Octave count	Specify a logarithmic (base 2) series of values, and the number of values to calculate in each octave.
Exponential count	Specify an exponential (base e) series of values, and the number of values to calculate.

2. If you selected **Single value**, type the value of the sweep definition in the **Value** box.

   If you selected another sweep type, do the following:

   a. Type the starting value of the variable range in the **Start** text box.
   b. Type the final value of the variable range in the **Stop** text box.

Warning	Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

3. If you selected **Linear step** as the sweep type, type the step size in the **Step** box.

   The step size is the difference between variable values in the sweep definition. The step size determines the number of design variations between the start and stop values. Maxwell will solve the model at each step in the specified range, including the start and stop values. The step size can be negative, when the **Stop** value is less than the **Start** value

   If you selected another sweep type, type the number of points, or variable values, in the sweep definition in the **Count** text box. For **Decade count** and **Octave count**, the **Count** value specifies the number of points to calculate in every decade or octave. For **Exponential count**, the **Count** value is the total number of points. The total number of points includes the start and stop values.

Related Topics

Synchronizing Variable Sweep Definitions

Synchronizing Variable Sweep Definitions

By default, variable sweep definitions are nested. Alternatively, you can synchronize the variable sweep definitions if they have the same number of sweep points.

For example, if you synchronize a sweep definition that includes values of 1, 2, and 3 inches with a second sweep definition that includes values of 4, 5, and 6 inches, Maxwell will solve 3 design

variations. The first variation is solved at the variable values of 1 and 4; the second variation is solved at the variable values 2 and 5; and the third variation is solved at the final variable values 3 and 6.

To synchronize variable sweep definitions:

1. Under the **Sweep Definitions** tab of the **Setup Sweep Analysis** dialog box, select the rows containing the sweep definitions you want to synchronize.
2. Click **Sync**.

   The synchronized sweeps are given a group number, which is listed in the **Sync #** column.

Optionally, view the design variations that are to be solved in table format under the **Table** tab.

Related Topics

Specifying Variable Values for a Sweep Definitions

Modifying a Variable Sweep Definition Manually

You can manually modify the variable values that are solved for a parametric setup by explicitly changing, adding, or deleting existing points in a variable sweep definition under the **Table** tab of the **Setup Sweep Analysis** dialog box.

To manually modify a variable sweep definition:

1. Click the **Table** tab of the **Setup Sweep Analysis** dialog box.

   The design variations Maxwell solves for the parametric setup listed in table format.
2. Do one of the following:
   - To modify a variable value, click a value text box in the table and type a new value.
   - To delete a variable value from the sweep definition, click the row you want to delete, and then click **Delete**.
   - To add a new variable value to the sweep definition, click **Add**. Then click in the value text box and type a new value.

Warning	Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

Your modifications are tracked and available for viewing at the bottom of the **Setup Sweep Analysis** dialog box under the **Sweep Definitions** tab. The operations you performed are listed with descriptions.

Warning	If you modify an original sweep definition using the **Add/Edit Sweep** dialog box after you have manually modified its table of design variations, your manual modifications become invalid and are removed. A warning is displayed to inform you that your manual values are about to become invalid, so you can decide whether or not to proceed.

Related Topics

Adding a Variable Sweep Definition

Overriding a Variable's Current Value in a Parametric Setup

Overriding a Variable's Current Value in a Parametric Setup

If you choose not to sweep a variable, Maxwell uses the variable's current value set for the nominal design when it solves the parametric setup. To override the current variable value for a parametric setup:

1. In the **Setup Sweep Analysis** dialog box, click the **General** tab.

   Under **Starting Point**, all of the current independent design variable values are listed.
2. Click the **Value** box of the variable with the value you want to override for the parametric setup.
3. Type a new value in the **Value** box, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered will be used for the parametric setup. For this parametric setup, the new value will override the current value in the nominal design.

Note	Alternatively, you can select the **Override** option first, and then type a new variable value in the **Value** box.

4. Optionally, click a new unit in the **Units** box.

To revert to the current variable value, clear the **Override** option.

Warning	Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

Related Topics

Adding a Variable Sweep Definition

Modifying a Variable Sweep Definition Manually

Specifying a Solution Setup for a Parametric Setup

To specify the solution setup that Maxwell analyzes when it solves a parametric setup:

1. In the **Setup Sweep Analysis** dialog box, click the **General** tab.
2. Select the solution setup you want Maxwell to use when it solves the parametric setup.

   Maxwell solves the parametric setup using the solution setup you select. If you select more than one, results are generated for all selected solution setups.

Related Topics

Specifying the Solution Quantity to Evaluate for Parametric Analysis
Specifying a Solution Quantity's Calculation Range

Specifying the Solution Quantity to Evaluate for Parametric Analysis

When you add a parametric setup, you can identify one or more solution quantities to be presented in the **Post Analysis Display** dialog box. The solution quantities are specified by mathematical expressions that are composed of basic quantities, such as output variables. When you view the results, Maxwell extracts the solution quantities and lists them in the results table.

1. In the **Setup Sweep Analysis** dialog box, click the **Calculations** tab.

   This displays a table that will show Solutions and associated Calculations. Below the table, are control buttons to **Setup Calculations...** and **Delete**.
2. Click **Setup Calculations**.

   This displays the **Add/Edit Calculation** dialog. The dialog contains panes to set the **Context**, the **Trace** tab for the **Calculation Expression**, and the **Calculation Range** tab for the **Calculation Range**.

   Follow the procedure to Setup Calculations for Optimetrics.
3. Click **Add Calculation** to add the expression in the **Add/Edit Calculation** dialog **Calculation Expression** field to the Calculations tab of the **Setup Sweep Analysis** dialog.
4. Click **Done** to close the **Add/Edit Calculation** dialog.

Related Topics

Specifying a Solution Quantity's Calculation Range
Specifying Output Variables
Setup Calculations for Optimetrics.

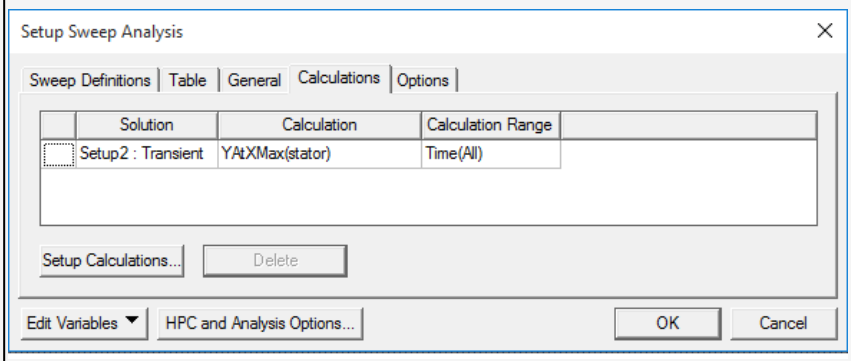
Setup Calculations for Optimetrics

The **Setup** dialogs for each of the Optimetrics types include a **Setup Calculations** button. Clicking this displays the **Add/Edit Calculation** dialog box. The dialog box contains distinct panes and tabs to set the **Context**, the **Calculation Expression**, and the **Calculation Range**.

The **Context** pane contains fields for the Report Type to use, the Solution, and depending on the Report Type selection, the Geometry.

The **Trace** tab contains fields for the Calculation expression, and, to build the expression, a Category list, a Quantity list with a Text Filter field, and a list of Functions available for the selected Category. The Range function button opens a dialog in which you can define a range function to apply a function to the expression.

The Category list for the **Trace** tab includes Variables and Output Variables. An Output Variables... button lets you open a dialog box to define and edit the Output Variables.

Note	When specifying field calculation for a solution that is defined with varying save field, select "All" time points as the Calculation Range and apply a range function to the Calculation as shown in the parametric setup example below.  Field calculation with geometry context is not supported for a solution that is defined with varying save field.

To setup an Optimetrics calculation:

1. Click the **Setup Calculations** button to open the **Add/Edit/Calculation** dialog.
2. In the **Report Type** text field in the **Context** pane, select from the drop down list of available types.

   Selecting Fields as the Report type causes the **Geometry** field to display.
3. In the **Solution** text box, select from the drop down list of available solutions.
4. If the **Geometry** field is available, select from the drop down list.
5. In the **Trace tab**, specify the solution Category, a Quantity, and Functions. The resulting expression will be displayed in the **Calculation Expression** field.
   a. Select the **Category** from the list.
      The selection appears in the **Calculation Expression** field, and the Quantity and Function fields list what is available for the corresponding selection.
   b. Select the **Quantity** from the list.
      The selected quantity appears in the **Calculation Expression** field.
      If the **Quantity** list is long, you can filter it for easier selection by typing in the text filter field. Only quantities that contain those alphanumeric characters anywhere in their name will remain visible in the list.
      If you want to create an output variable that represents the solution quantity, do the following:
      - Click the **Output Variables** button.

        The **Output Variables** dialog box appears.
      - Add the expression you want to evaluate, and then click **Done**.

        The recently created output variable appears in the **Quantity list**.

- Click a new output variable in the Quantity pull-down list.

Note	The calculation you specify must be able to be evaluated into a single, real number.

The selected Quantity appears in the **Calculation Expression** field.

c. Select the **Function** from the list.
The selected function is applied to the **Quantity** in the **Calculation Expression** field.

6. To apply a **Range function** to the **Calculation Expression**, see Setting a Range function.
7. Click **Add Calculation** to add the expression in the **Add/Edit Calculation** dialog **Calculation Expression** field to the Calculations tab of the **Setup Sweep Analysis** dialog.
8. Click **Done** to close the **Add/Edit Calculation** dialog box.

Related Topics

Specifying a Solution Quantity to Evaluate

Setting a Range Function

Specifying a Solution Quantity's Calculation Range

The calculation range of a solution quantity determines the value of intrinsic variables such as frequency (F) at which the solution quantity will be extracted. For a parametric setup, the calculation range must be a single value.

1. In the **Setup Sweep Analysis** dialog box, click the **Calculations** tab.
2. Click the **Setup Calculations** button.
The **Add/Edit/Calculation** dialog box appears.
3. Select the **Calculation Range** tab.
4. In the **Variable** list, click an intrinsic variable.
Depending on the variable and Report Type, either a single value or All appears in the Value field.
5. To change the Value, click the ellipsis button.to display a dialog box that lets you select particular values, or specify a range of values (for Time sweeps). By default, **Use all values** is selected.
   - For variables other than Time, unchecking **Use all values** enables a list of available values. You can use the buttons to **Clear All** or **Select All** values.
     a. Select the **SweepDefault** or **Edited** radio buttons to specify whether to accept the default or edited sweeps.
     b. To edit the sweeps further, select the ellipsis [**...**] button to display an **Edit Sweep** dialog box. For frequency variables, this lets you specify a single value, linear step, linear count, decade count, octave count, or exponential count. You can **Add** legal values to the list of sweep values, **Update** the list for changes, or **Delete** selected entries.

c. When finished selecting/editing sweep values, close the dialog box. The values you select are displayed in the Value field.

- For Time sweeps:
  a. Select the **Specify range** radio button to enable the **Min** and **Max** value fields.
  b. Enter the desired **Min** and **Max** values, and choose the associated units of measure from the drop down lists.
  c. When finished, close the dialog box. The values you selected are displayed in the Value field.

6. Click **Update**, and then click **Done**.

Using Distributed Analysis

If you have purchased the appropriate license, Maxwell supports distributed solve, which involves distributing rows of a parametric table during Optimetrics solve.

If you do a distributed solve, Maxwell launches solver engines on multiple machines, assuming that you have configured your machines appropriately.

To run a distributed analysis:

1. Under **Optimetrics** in the project tree, right-click the specific parametric setup.
   A shortcut menu appears.
2. Select **Distribute Analysis** from the shortcut menu.

Note	After you define a parametric sweep, a shortcut menu becomes available when you right-click the setup name.

   While the analysis is running, you can access parent and child progress bars. By default, only the main progress bar is displayed, while the child progress bars (or subtasks) remain hidden. You can toggle between showing and hiding the child progress bars.

   To show the child progress bars:

   - Right-click the progress window, and select **Show Subtask Progress Bars**.

   To hide the child progress bars:

   - Right-click the progress window, and select **Hide Subtask Progress Bars**.

Related Topics

General Options: Remote Analysis
Solving Remotely (Windows Only)
Setting Up Distributed Analysis with Licensing
Viewing Distributed Analysis Subtasks

Adding a Parametric Sweep from a File

You can specify the parameters for a parametric sweep in a spreadsheet that uses either a .csv (comma delimited) or .txt (tab delimited) format. You can then import the parametric sweep using

the **Maxwell (2D or 3D)>Optimetrics Analysis>Add Parametric from File** command.

For example, a .txt spreadsheet file could resemble the following:

```
a $b $c[in] d[m] $e $f
0.1 mil 2mm 11 21 0.6in 8
0.2mil 3mm 1.3 2.6mm 3 9cm
...
```

The first row lists the Project and Design Variable names, and when followed by parentheses, the units. The following rows provide the variable values and units. Project or Design variables must be defined before they are accepted from a file. The characters in variable names are not case sensitive. Consecutive separators are treated as one separator.

The header row also takes units in ( ) as well as the conventional [ ].

Related Topics

Setting up a Parametric Analysis

Optimization Overview

Optimetrics interfaces with Ansys Electromagnetics products to enable the optimization of a wide variety of design parameters based on variable geometry, materials, excitations, component values, etc. Optimization is the process of locating the minimum of a user-defined cost function. Optimetrics modifies the variable values until the minimum is reached with acceptable accuracy.

Related Topics

Setting Up an Optimization Analysis

Choosing an Optimizer

Choosing an Optimizer

Conducting an optimization analysis allows you to determine an optimum solution for your problem. In Maxwell optimization analyses, you have five choices of optimizer, though in most cases, the Sequential Nonlinear Programming optimizer is recommended.

- Sequential Nonlinear Programming (SNLP)
- Sequential Mixed Integer NonLinear Programming (SMINLP)
- Quasi Newton
- Pattern Search
- Genetic Algorithm
- MATLAB Optimizer

Additional Optimizers: These use a Decision Support Process (DSP) based on satisfying criteria as applied to the parameter attributes using a weighted aggregate method. In effect, the DSP can

be viewed as a postprocessing action on the Pareto fronts as generated from the results of the various optimization methods.

- Screening (Search based): This is a non-iterative direct sampling method that uses a quasi-random number generator based on the Hammersley algorithm. You can start with Screening to locate the multiple tentative optima and then refine with NLPQL or MISQP to zoom in on the individual local maximum or minimum value. Usually Screening is used for preliminary design, which can lead you to apply one of the other approaches for more refined optimization results.
- Multi-Objective Genetic Algorithm: This is an iterative random search algorithm that can optimize problems with continuous input parameters. It is better for calculating the global optima. You can start with MOGA to locate the multiple tentative optima and then refine with NLPQL or MISQP to zoom in on the individual local maximum or minimum value.
- Nonliner Programming by Quadratic Lagrangian (Gradient): This is a gradient-based, single-objective optimizer based on quasi-Newton methods. Ideally suited for local optimization.
- Mixed-Integer Sequential Quadratic Programming (Gradient and Discrete): This is a gradient-based, single-objective optimizer that solves mixed-integer non-linear programming problems by a modified sequential quadratic programming (SQP) method. Ideally suited for local optimization.
- Adaptive Multiple Objective (Gradient): This is an iterative, multi-objective optimizer that employs a Kriging response surface and MOGA. In this method, the use of a Kriging response surface allows for a more rapid optimization process because all design points are not evaluated except when necessary and part of the population is simulated by evaluations of the Kriging response surface, which is constructed of all design points submitted by Multi-Objective Genetic Algorithm (MOGA).
- Adaptive Single Objective (Gradient): This is a gradient-based, single-objective optimizer that employs an OSF (Optimal Space-Filling) DOE, a Kriging response surface, and MISQP.

The optimizers assume that the nominal problem you are analyzing is close to the optimal solution; therefore, you must specify a domain that contains the region in which you expect to reach the optimum value.

All optimizers allow you to define a maximum limit to the number of iterations to be executed. This prevents you from consuming your remaining computing resources and allows you to analyze the obtained solutions. From this reduced range, you can further narrow the domain of the problem and regenerate the solutions.

All optimizers also allow you to enter a coefficient in the **Add Constraints** window to define the linear relationship between the selected variables and the entered constraint value. For the SNLP and NMINLP optimizers, the relationship can be linear or nonlinear. For the Quasi Newton and Pattern Search optimizers, the relationship must be linear.

Cost functions can be quite nonlinear. As a result, during the function evaluations of the algorithm, the cost function can vary significantly. Also, it is important to understand the relationship between optimization function evaluation and iteration. Every iteration, depending on the number of parameters to be optimized, performs several function evaluations. These function evaluations, depending on how nonlinear the cost function is, could show drastic changes. The presence of drastic changes has no bearing on whether the optimization algorithm converged or not.

In the case of non-gradient search-based optimization algorithms, such as "pattern search," which are entirely based on function evaluations, one could see drastic changes in the function evaluations depending on how nonlinear the cost function is. This could seem misleading as if the algorithm did not converge since in theory one expects the cost function to decrease from one iteration to the next. The optimetrics, however, reports function evaluations and not necessarily the optimizer performance per iteration.

Note	The MATLAB optimizer displays function evaluation when the **Show all functions evaluation** check box is selected. If the check box is not selected, it displays iteration.

Quasi Newton

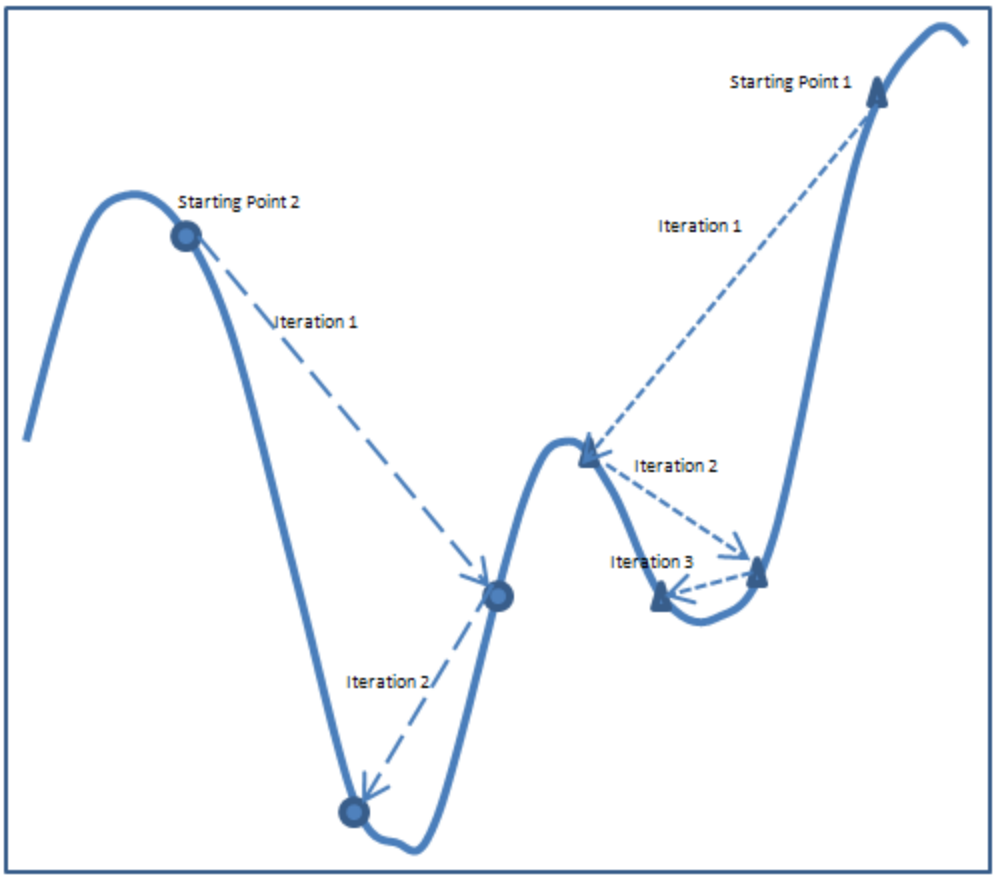
If the Sequential Non Linear Programming Optimizer has difficulty, and if the numerical noise is insignificant during the solution process, use the Quasi Newton optimizer to obtain the results. The Quasi Newton optimizer works on the basis of finding a minimum or maximum of a cost function which relates variables in the model or circuit to overall simulation goals. The user defines one or more variables in the problem definition and a cost function in the optimization setup. The cost function relates the variable values to field quantities, design parameters like force or torque, power loss, etc. The optimizer can then maximize or minimize the value of the design parameter by varying the problem variables.

Sir Isaac Newton first showed that the maximum or minimum of any function can be determined by setting the derivative of a function with respect to a variable (x) to zero and solving for the variable. This approach leads to the exact solution for quadratic functions. However, for higher order functions or numerical analysis, an iterative approach is commonly taken. The function is approximately locally by a quadratic and the approximation is solved for the value of x. This value is placed back into the original function and used to calculate a gradient which provides a step direction and size for determining the next best value of x in the iteration process.

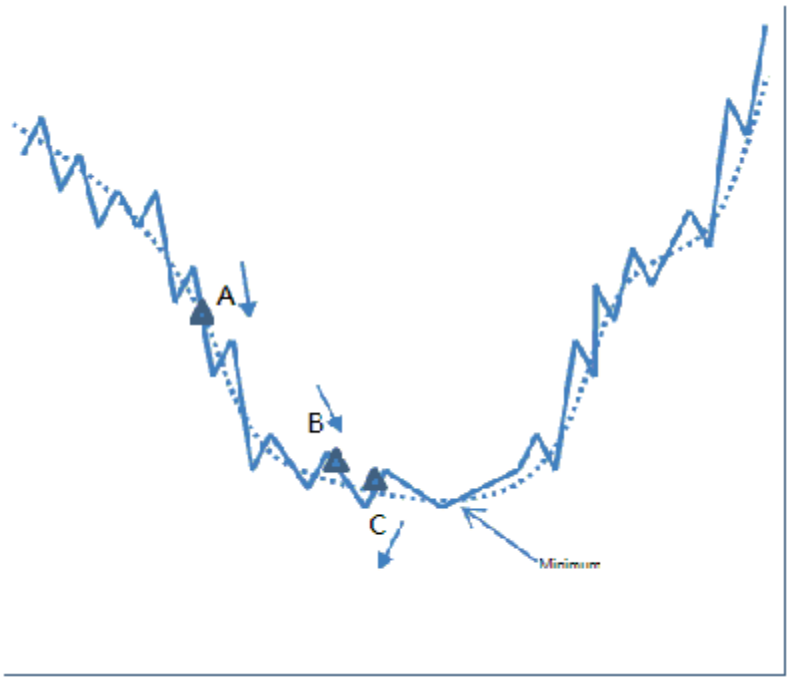
In the Quasi-Newton optimization procedure, the gradients (Hessian) are not well behaved functions and are calculated numerically. Essentially, the change in the estimate of x and the change in the gradient are used to estimate the Hessian for the next iteration. The ratio of the change in the gradients to the change in the values of x provides the Hessian for the next step and is know as the quasi-Newton condition. In order to perform the Quasi-Newton optimization, at least three solutions are required for each parameter being varied. This can have a significant computational cost depending upon the type of analysis being performed.

There are numerous methods described in the literature for solving for the Hessian and the details of the method used by Optimetrics are beyond the scope of this document. However, as the Quasi-Newton method is, at its heart, a gradient method, it suffers from two fundamental problems common to optimization. The first is the possible presence of local minima. The following figure illustrates the problem of local minima. In this scenario, you can see that in order to find the minimum of the function over the domain, a number of factors will determine the overall success including the initial starting point, the initial set of gradients calculated, the allowable step size, etc. Once the optimizer has located a minimum, the Quasi-Newton approach will locate the bottom and will not search further for other possible minima. In the example shown, when the optimizer

begins at the point labeled "Starting Point 1" the minima it finds is a local minima and not a good global solution to the problem.



The second basic issue with Quasi-Newton optimization is numerical noise. In gradient optimization, the derivatives are assumed to be smooth, well behaved functions. However, when the gradients are calculated numerically, the calculation involves taking the differences of numbers that get progressively smaller. At some point, the numerical imprecision in the parameter calculations becomes greater than the differences calculated in the gradients and the solution will oscillate and may never reach convergence. To illustrate this, consider the figure shown below. In this scenario, the optimizer is looking for the point labeled "minimum". Three possible solutions are labeled A, B and C, with each arrow indicating the direction of the derivative of the function at that point. If points A and B represent the last two solution points for the parameter, then it is easy to see that the changes in the magnitude and the consistent direction of the derivatives will serve to push the solution closer to the desired minimum. If, however, points A and C are the last two solution points respectively, the magnitude indicates the proper direction of movement, but the derivatives are opposite, possibly causing the solution to move away from the minimum, back in the direction of point A.



In order to use the Quasi-Newton optimizer effectively, the cost function should be based on parameters that exhibit a smooth characteristic (little numerical noise) and a starting point of the optimization should be chosen somewhat close to the expected minimum based on an understanding of the physical problem being optimized. This becomes increasingly difficult, however, when multiple parameters are being varied or when multiple parameters are to be optimized. In addition, the computational burden of multivariate optimization with Quasi-Newton increases geometrically with the number of variables being optimized. As a result, this method should only be attempted when 1 or 2 variables are being optimized as a time.

For more information regarding Quasi-Newton optimization methods, see the following reference:

Schoenberg, Ronald. *Optimization with the Quasi-Newton Method*. Aptech Systems, Inc. 2001.

Related Topics

Optimization Setup for Quasi Newton Optimizer

Pattern Search

If the noise is significant in the nominal project, use the Pattern Search optimizer to obtain the results. It performs a grid-based simplex search, which makes use of simplices: triangles in 2D space or tetrahedra in 3D space. A simplex is a Euclidean geometric spatial element having the

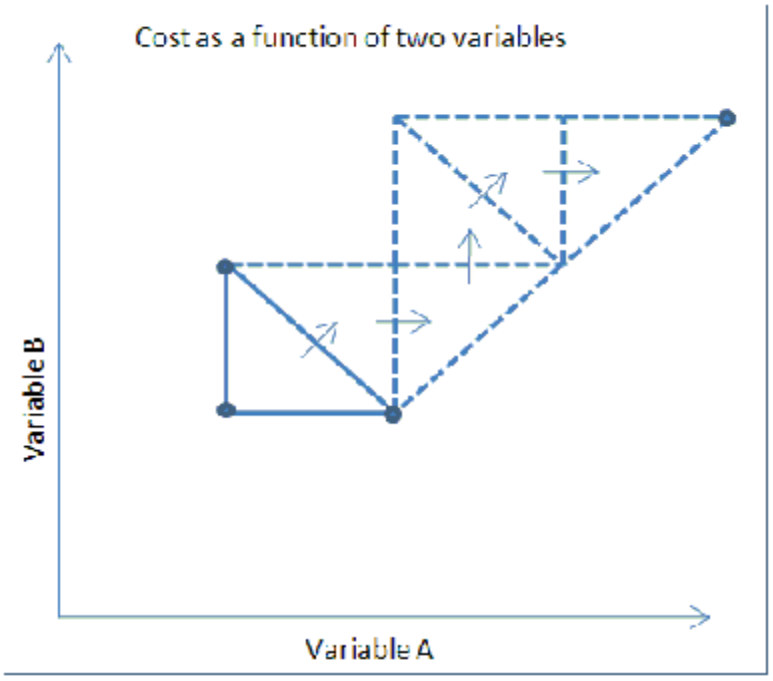
minimum number of boundary points, such as a line segment in one-dimensional space, a triangle in two-dimensional space, or a tetrahedron in three-dimensional space.

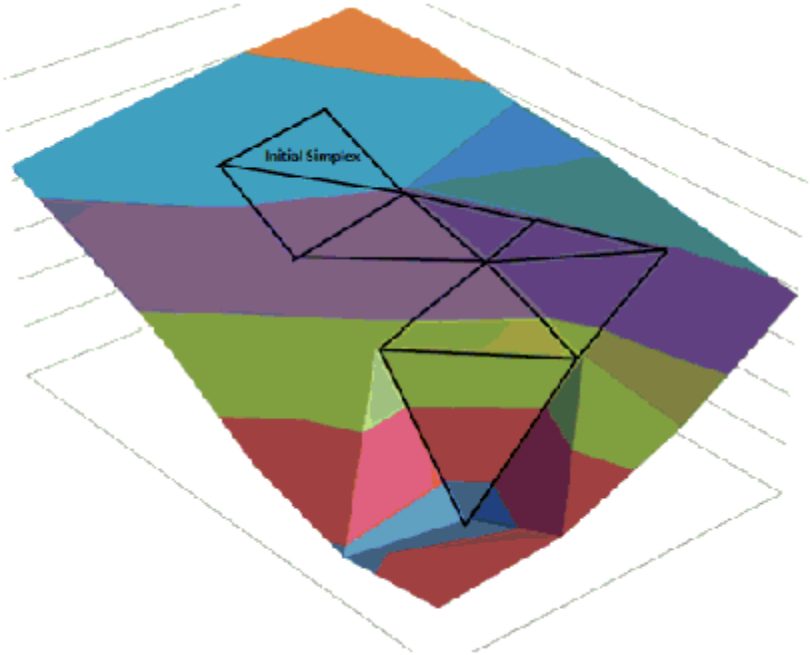
The cost value is calculated at the vertices of the simplex. The optimizer mirrors the simplex across one of its faces based on mathematical guidelines and determines if the new simplex provides better results. If it does not produce a better result, the next face is used for mirroring and the pattern continues. If no improvement occurs, the grid is refined. If improvement occurs, the step is accepted and the new simplex is generated to replace the original one. The figures below illustrates a triangular simplex mirrored several times to demonstrate the pattern search approach in two variables and the simplices superimposed on a 2D cost function to demonstrate the convergence toward a minimum in the cost function.

Cost functions can be quite nonlinear. As a result, during the function evaluations of the algorithm, the cost function can vary significantly. Also, it is important to understand the relationship between optimization function evaluation and iteration. Every iteration, depending on the number of parameters to be optimized, performs several function evaluations. These function evaluations, depending on how nonlinear the cost function is, could show drastic changes. The presence of drastic changes has no bearing on whether the optimization algorithm converged or not.

In the case of non-gradient search-based optimization algorithms, such as "pattern search," which are entirely based on function evaluations, one could see drastic changes in the function evaluations depending on how nonlinear the cost function is. This could seem misleading as if the algorithm did not converge since in theory one expects the cost function to decrease from one iteration to the next. The optimetrics, however, reports function evaluations and not necessarily the optimizer performance per iteration.

Note	The MATLAB optimizer displays function evaluation when the **Show all functions evaluation** check box is selected. If the check box is not selected, it displays iteration.





The Pattern Search algorithms are extensible to three variable optimization by using tetrahedral simplices, however, they are not easily represented in graphical form. Generally, Pattern Search algorithms are not used when more than three variables are used in the optimization.

When there is not improvement in the cost function regardless of the direction the simplex is mirrored, then the simplex is subdivided into smaller simplices and the process restarted.

Pattern Search algorithms have several advantages over Quasi-Newton algorithms. First, they are less sensitive to noise because the cost function is evaluated at all node points on the simplex and the numerical noise averages out over the simplex. The second advantage is that the number of initial solutions is generally smaller as shown in the table. However, since the pattern search does not use gradient information to locate the minimum the process converges more slowly toward the true minimum, taking more steps to successively divide the simplices as the minimum is approached.

Related Topics

Optimization Setup for Pattern Search Optimizer

Sequential Non-linear Programming (SNLP)

The main advantage of SNLP over Quasi Newton is that it handles the optimization problem in more depth. This optimizer assumes that the optimization variables span a continuous space. As a result, there is no Minimum Step Size specified in this optimizer and the variables may take any value within the allowable constraints and within the numerical precision limits of the simulator. Like Quasi Newton, the SNLP optimizer assumes that the noise is not significant. It does reduce the effect of the noise, but the noise filtering is not strong.

The SNLP optimizer approximates the FEA characterization with Response Surfaces (RS). With the FEA-approximation and with light evaluation of the cost function, SNLP has a good approximation of the cost function in terms of the optimization variables. This approximation allows the SNLP optimizer to estimate the location of improving points. The overall cost approximations are more accurate. This allows the SNLP optimizer a faster practical convergence speed than that of quasi Newton.

The SNLP Optimizer creates the response surface using a Taylor Series approximation from the FEA simulation results available from past solutions. The response surface is most accurate in the local vicinity. The response surface is used in the optimization loop to determine the gradients and calculate the next step direction and distance. The response surface acts as a surrogate for the FEA simulation, reducing the number of FEA simulations required and greatly speeding the problem. Convergence improves as more FEA solutions are created and the response surface approximation improves.

The SNLP method is similar to the Sequential Quadratic Programming (SQP) method in two ways: Both are sequential, updating the optimizer state to the current optimal values and iterating. Sequential optimization can be thought of a walking a path, step by step, toward an optimal goal. SNLP and SQP optimizers are also similar in that both use local and inexpensive surrogates.

However, in the SNLP case, the surrogate can be of a higher order and is more generally constrained. The goal is to achieve a surrogate model that is accurate enough on a wider scale, so that the search procedures are well lead by the surrogate, even for relatively large steps. All functions calculated by the supporting finite element product (for example, Maxwell 3D or HFSS) is assumed to be expensive, while the rest of the cost calculation (for example, an extra user-defined expression) — which is implemented in Optimetrics — is assumed to be inexpensive. For this reason, it makes sense to remove inexpensive evaluations from the finite element problem and, instead, implement them in Optimetrics. This optimizer holds several advantages over the Quasi Newton and Pattern Search optimizers.

Most importantly, due to the separation of expensive and inexpensive evaluations in the cost calculation, the SNLP optimizer is more tightly integrated with the supporting FEA tools. This tight integration provides more insight into the optimization problem, resulting in a significantly faster optimization process. A second advantage is that the SNLP optimizer does not require cost-derivatives to be approximated, protecting against uncertainties (noise) in cost evaluations. In addition to derivative-free state of the RS-based SNLP, the RS technique also proves to have noise suppression properties.

Related Topics

Optimization Setup for SNLP Optimizer

Sequential Mixed Integer NonLinear Programming

The Sequential Mixed Integer Nonlinear Programming (SMINLP) optimizer is equivalent to the SNLP optimizer with only one difference. Many problems require variables take only discrete values. One example might be to optimize on the number of turns in a coil. To be able to optimize on number of turns or quarter turns, the optimizer must handle discrete optimization variables. The SMINLP optimizer can mix continuous variables among the integers, or can have only integers, and works if all variables are continuous. The setup resembles the setup for SNLP, except that you must flag the integer variables.supporting integer variables. You can set up internal variables based on the integer optimization variable.

For example, consider N to be an integer optimization variable. By definition it can only assume integer values. You can establish another variable, which further depends on this one: K = 2.345 * N, or K = sin(30 * N). This way K has a discrete value, but is not necessarily integer. Or, one can use N directly as a design parameter.

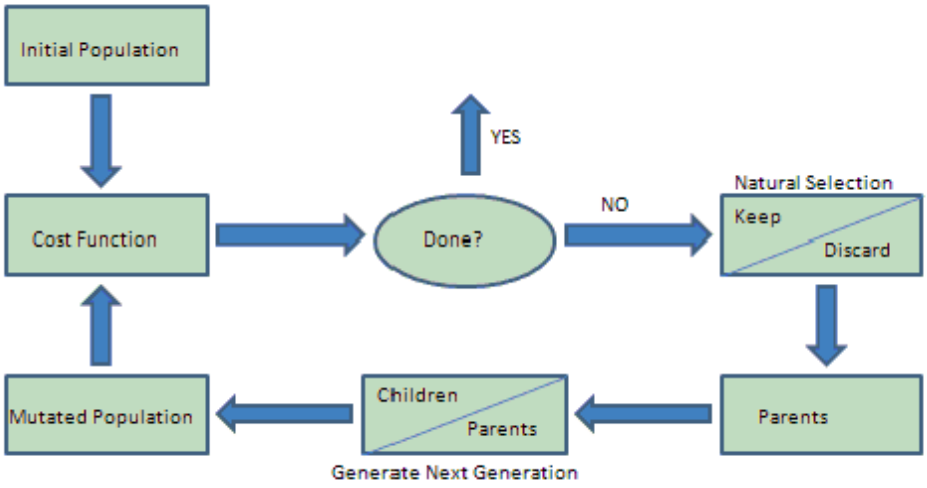
Related Topics

Optimization Setup for SMINLP Optimizer

Genetic Algorithm

Genetic Algorithm (GA) optimizers are part of a class of optimization techniques called stochastic optimizers. They do not use the information from the experiment or the cost function to determine where to further explore the design space. Instead, they use a type of random selection and apply it in a structured manner. The random selection of evaluations to proceed to the next generation has the advantage of allowing the optimizer to jump out of a local minima at the expense of many random solutions which do not provide improvement toward the optimization goal. As a result, the GA optimizer will run many more iterations and may be prohibitively slow.

The Genetic Algorithm search is an iterative process that goes through a number of generations (see picture below). In each generation some new individuals (Children / Number of Individuals) are created and the grown population participates in a selection (natural-selection) process that in turn reduces the size of the population to a desired level (Next Generation / Number of Individuals).



When a smaller set of individuals must be created from a bigger set, the GA selects individuals from the original set. During this process, better fit (in relation to the cost function) individuals are preferred. In the elitist selection, simply the best so many individuals are selected, but if you turn on the roulette selection, then the selection process gets relaxed. An iterative process starts selecting the individuals and fill up the resulting set, but instead of selecting the best so many, we use a roulette wheel that has for each selection-candidate divisions made proportional to the fitness level (relative to the cost function) of the candidate. This means that the fitter the individual is, the larger the probability of his survival will be.

Related Topics

Optimization Setup for Genetic Algorithm Optimizer

Optimization Variables in Design Space

Cost Function

Advanced Genetic Algorithm Optimizer Options

MATLAB Optimizer

The MATLAB optimizer option lets you pass a script to MATLAB to perform the optimization. When the optimization is analyzed, MATLAB is launched and a script is passed in to MATLAB to perform the optimization. During the optimization, MATLAB will call back into our application to perform the solve and compute the cost. The cost will be reported back to MATLAB, and MATLAB's optimization will determine the next step in the optimization.

The optimization script is specified as part of the optimization setup. By modifying the optimization script, users can change the optimization parameters and optimization method as well as use the full power of MATLAB in their optimization.

Running the Optimization

The MATLAB optimization is launched just like any other optimization. The Message Window will display status messages when MATLAB is being launched, and status messages will be generated for each solve that is being performed.

In most cases, MATLAB will terminate when the optimization has been completed. Some reasons why MATLAB would not terminate are:

- The user has modified the MATLAB script to not terminate MATLAB after the optimization.
- A syntax error or some other has occurred.
- The user has added some other code which runs after the optimization has completed.

System Requirements

In order to use MATLAB to perform optimizations from your application:

- A version of MATLAB must be installed on your system.
- The computing platform (i.e. 64 bit or Linux) of MATLAB MUST match the platform of the Ansys application you are using it with.
- You must have the MATLAB Optimization Toolkit installed.

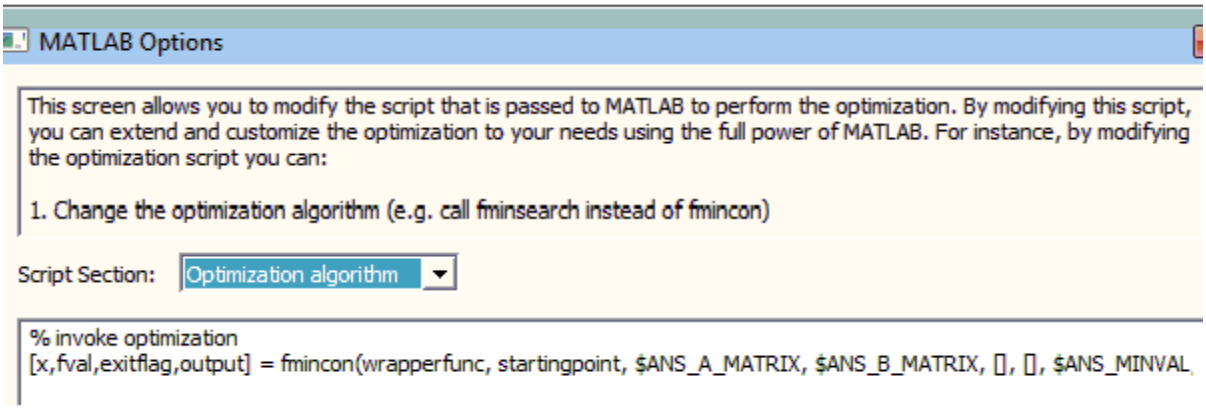
Specifying the MATLAB Location

The Miscellaneous options panel contains a setting for the MATLAB application location. This setting must to point to the version of MATLAB to be used for performing the optimization. The platform (i.e. 64 bit or Linux) of the specified version of MATLAB must match the platform of this application.

MATLAB Optimization Setup

MATLAB optimization starts by creating an optimization and selecting MATLAB from the optimizer dropdown list. If you select MATLAB as the optimizer, the **Setup Optimization** dialog displays a **Setup...** button.

Select **Setup...** to open the **MATLAB Options** dialog.



Thee upper text panel is informative. The Script section drop down lets you select a lower panel display for Optimization algorithm, Options, or the Full script template.

This panel allows you to modify the script that is passed to MATLAB to perform the optimization. The complete script contains all the instructions necessary for MATLAB to connect to Maxwell and perform the optimization. The drop-down selection lets you view only the portion of code of interest without having to view the full script. The choices are:

- **Optimization algorithm**: displays only the line of code invoking the actual optimization function. By changing this line, the user can use a different MATLAB function for optimization. By default we use `fmincon()`, which is a derivative-based constrained optimization. By modifying this line, the user could replace the `fmincon()` call with `fminsearch()` to use an unconstrained pattern searching optimizer or another optimization function. *See the MATLAB documentation for details about available optimization functions.*
- **Options**: Each optimization function contains many options and parameters, which are set in the MATLAB script prior to actually calling the optimization function. By modifying these options, the optimization can be customized as desired. For instance, options can be set for `fmincon()` to specify the algorithm that it uses internally. *See the MATLAB documentation for details about options available for each optimization function.*
- **Full script template**: This choice displays the full optimization script that is passed to MATLAB.

The initial Script Section display for the Optimization algorithm shows the following:

```
% invoke optimization
[x,fval,exitflag,output] = fmincon(wrapperfunc, startingpoint,
[], [], [], [], $ANS_MINVAL, $ANS_MAXVAL, nlcon, options)
```

The initial Script Section Options display shows the following:

```
% customers can add their own options below
options = optimset(options, 'display', 'iter')
options = optimset(options, 'Algorithm', 'interior-point')
% options = optimset(options, 'PlotFcns', @optimplotfval)
```

You can modify the script to extend and customize the optimization to your needs. You must ensure that the script follows MATLAB syntax. For instance, by modifying the optimization script you can:

- Change the optimization algorithm (e.g. call fminsearch instead of fmincon)
- Change the parameters/options of the optimization algorithm (see the MATLAB documentation for details).
- Specify a plot function to provide graphical output during optimization.
- Specify a user defined output function to be called at completion or per iteration.

Symbols:

When modifying the MATLAB code, users can use symbols to represent values from the optimization setup. The symbols and their definitions are listed below.

$ANS_VARIABLE_LIST:	list of variables we are optimizing
$ANS_STARTING_POINT:	vector of starting values of variables used in the optimization
$ANS_MAXITERATIONS:	maximum number of iterations specified in optimization setup
$ANS_MINVAL:	vector of minimum values from optimization setup
$ANS_MAXVAL:	vector of maximum values from optimization setup
$ANS_MINSTEP:	vector of minimum step sizes from optimization setup
$ANS_MAXSTEP:	vector of maximum step sizes from optimization setup
$ANS_A_MATRIX	matrix of linear constraint coefficients (left-hand side) generated from optimization setup
$ANS_B_MATRIX	matrix of linear constraint bounds (right-hand side) generated from optimization setup

Note	The linear constraints as generated for MATLAB have the form [A][x] <= [B], where [A] is the coefficient matrix, [x] is the variable list matrix (column vector), and [B] is the bounds matrix (column vector).

Note	While modifying the script, please ensure that the script follows MATLAB syntax.

MATLAB Optimization Script Template

The script template shown in the Script Section is as follows:

```
% make sure platform matches
if strcmp(computer, '$ANS_EXPECTED_PLATFORM') ~= 1
 h = msgbox('32/64 platform does not match calling application,
exiting')
 uiwait(h)
 exit
end
% add installation dir to search path so .mex file can be found
originalpath = addpath('$ANS_EXEDIR')
% connect back to opticomengine
callbackinterface = optimex('connect', '$ANS_CONNECTIONSTRING')
% set up optimization
% variables are: $ANS_VARIABLELIST
startingpoint = $ANS_STARTINGPOINT
options = optimset('MaxIter', $ANS_MAXITERATIONS)
iterationCallbackWrapper = @(x, optimValues, state) optimex
('notifyiterationcomplete', callbackinterface, x, optimValues.fval,
state)
options = optimset(options, 'OutputFcn', iterationCallbackWrapper)
% halt execution so debugger can be attached
% h = msgbox('attach debugger if desired')
% uiwait(h)
% attributes that user can pass to optimization algorithm
% variables are: $ANS_VARIABLELIST
% this is the objective function which returns cost
wrapperfunc = @(x)optimex('eval', callbackinterface, x)
% this is our non linear constraint function, returns no constraints
returnempty = @(x)[];
nlcon = @(x) deal(returnempty(x), returnempty(x));
% DO NOT EDIT THIS LINE - START OPTIONS SECTION
% customers can add their own options below
options = optimset(options, 'display', 'iter')
options = optimset(options, 'Algorithm', 'interior-point')
% options = optimset(options, 'PlotFcns', @optimplotfval)
% DO NOT EDIT THIS LINE - END OPTIONS SECTION
```

```
% DO NOT EDIT THIS LINE - START OPTIMIZATION ALGO SECTION
% invoke optimization
[x,fval,exitflag,output] = fmincon(wrapperfunc, startingpoint, $ANS_A_
MATRIX, $ANS_B_MATRIX, [], [], $ANS_MINVAL, $ANS_MAXVAL, nlcon,
options)
% DO NOT EDIT THIS LINE - END OPTIMIZATION ALGO SECTION
% write exit message to Ansoft message window
(warning=0,error=1,info=2)
optimex('postansoftmessage', callbackinterface, 2, output.message)
% notify opticomengine that optimization is finished
optimex('optimizationfinished', callbackinterface, exitflag)
% restore original path
path = originalpath
% note: comment below line if you want MATLAB to remain
% running after optimization
exit
```

Related Topics

Optimization Setup for the MATLAB Optimizer

Tools>General Options:Miscellaneous tab

Screening (Shifted Hammersley Sampling) Optimization

The Screening (Shifted Hammersley Sampling) algorithm is used for sample generation by all methods except NLPQL. The conventional Hammersley sampling algorithm is a quasi-random number generator, which has very low discrepancy and is used for quasi-Monte Carlo simulations.

A low-discrepancy sequence is defined as a sequence of points that approximate the equidistribution in a multi-dimensional cube in an optimal way. This means that the design space is populated almost uniformly by these sequences and that dimensionality is not a problem because of the inherent properties of Monte Carlo sampling. The number of points does not increase exponentially with an increase in the number of input parameters.

The conventional Hammersley sampling algorithm is constructed by using the radical inverse function. Any integer n can be represented as a sequence of digits n_0, n_1, n_2,..., n_m by the following equation:

$$n = n_0 n_1 n_2 n_3 \cdots n_m$$

For example, consider the integer 687459, which can be represented this way as n_0=6, n_1=8, and so on. Because this integer is represented with radix 10, you can write it as 687459=9+5*10+4*100 and so on. In general, for a radix *R* representation, the equation is:

$$n=n_m+n_{m-1}*R+\cdots+n_0$$

The inverse radical function is defined as the function that generates a fraction in (0, 1) by reversing the order of the digits in the previous equation about the decimal point, as shown.

$$\Phi_R(n)=0\,n_m\,n_{m-1}\,n_{m-2}\cdots n_0$$
$$=n_m*R^{-1}+n_{m-1}*R^{-2}+\cdots+n_0*R^{-(m-1)}$$

Thus, for a *k*-dimensional search space, the Hammersley points are given by the following expression:

$$H_k(i)=[i/N,\Phi_{R1}(i),\Phi_{R2}(i),\cdots,\Phi_{Rk-1}(i)]$$

Where *i*=0, ..., *N* indicates the sample points. Now, from the plot of these points, it is seen that the first row (corresponding to the first sample point) of the Hammersley matrix is zero and the last row is not 1. This implies that, for the *k*-dimensional hypercube, the Hammersley sampler generates a block of points that are skewed more toward the origin of the cube and away from the far edges and faces. To compensate for this bias, a point-shifting process is proposed that shifts all Hammersley points by the amount that follows:

$$\Delta=\frac{1}{2}N$$

This moves the point set more toward the center of the search space and avoids unnecessary bias. Thus, the initial population always provides unbiased, low-discrepancy coverage of the search space.

Related Topics

Optimization Variables in Design Space

Cost Function

Multi-Objective Genetic Algorithm (MOGA)

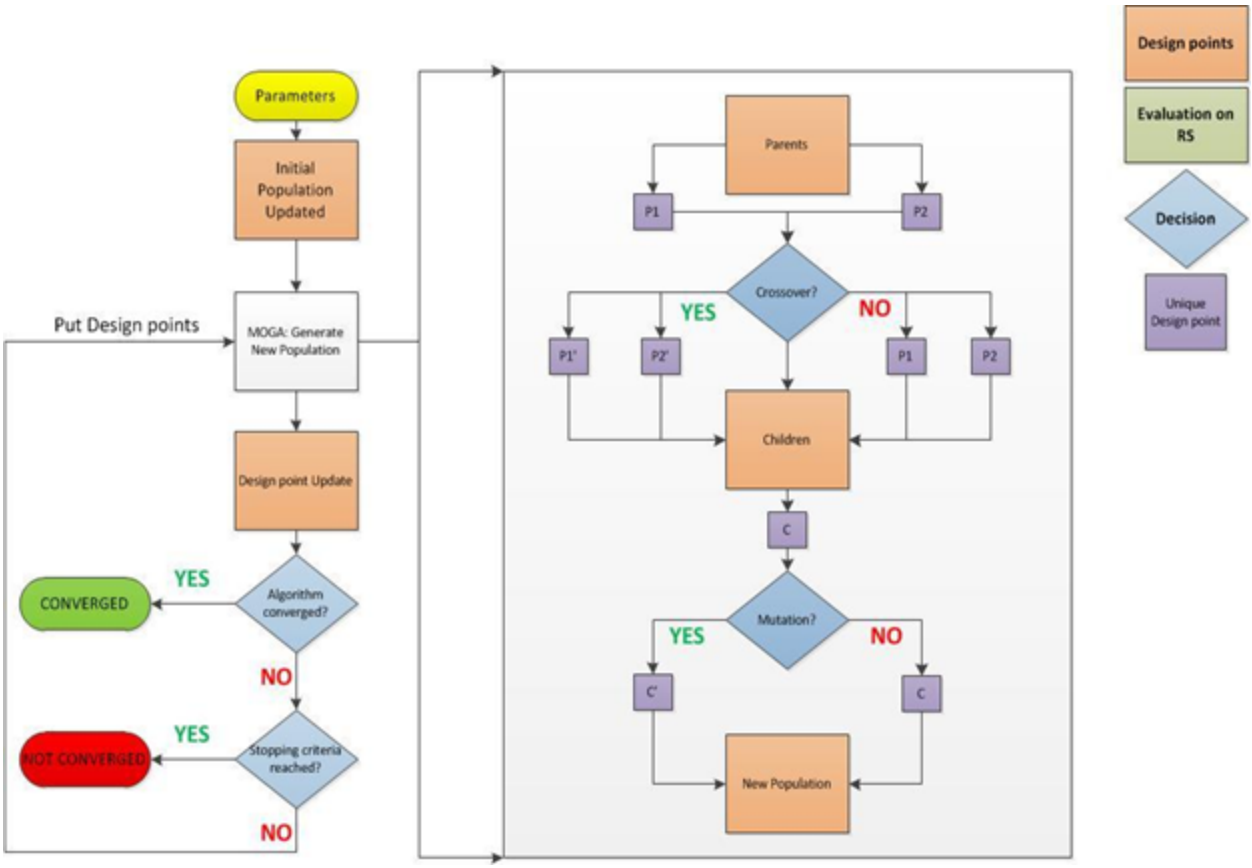
The Multi-Objective Genetic Algorithm (MOGA) used in GDO is a hybrid variant of the popular NSGA-II (Non-dominated Sorted Genetic Algorithm-II) based on controlled elitism concepts. It supports all types of input parameters. The Pareto ranking scheme is done by a fast, non-

dominated sorting method that is an order of magnitude faster than traditional Pareto ranking methods. The constraint handling uses the same non-dominance principle as the objectives. Therefore, penalty functions and Lagrange multipliers are not needed. This also ensures that the feasible solutions are always ranked higher than the infeasible solutions.

The first Pareto front solutions are archived in a separate sample set internally and are distinct from the evolving sample set. This ensures minimal disruption of Pareto front patterns already available from earlier iterations. You can control the selection pressure (and, consequently, the elitism of the process) to avoid premature convergence by altering the Maximum Allowable Pareto Percentage property. For more information about this and other MOGA properties, see Setup Multi-Objective Genetic Algorithm.

MOGA Workflow

The MOGO workflow follows:



MOGA Steps

1. **First Population of MOGA**

   The initial population is used to run MOGA.

2. **MOGA Generates a New Population**

MOGA is run and generates a new population via cross-over and mutation. After the first iteration, each population is run when it reaches the number of samples defined by the Number of Samples Per Iteration property. For details, see **MOGA Steps to Generate New Population** below.

3. **Design Point Update**

   The design points in the new population are updated.

4. **Convergence Validation**

   The optimization is validated for convergence.

   - **Yes: Optimization Converged**

     MOGA converges when the Maximum Allowable Pareto Percentage or the Convergence Stability Percentage has been reached.

   - **No: Optimization Not Converged**

     If the optimization is not converged, the process continues to the next step.

5. **Stopping Criteria Validation**

   If the optimization has not converged, it is validated for fulfillment of stopping criteria.

   - **Yes: Stopping Criteria Met**

     When the Maximum Number of Iterations criterion is met, the process is stopped without having reached convergence.

   - **No: Stopping Criteria Not Met**

     If the stopping criteria have not been met, MOGA is run again to generate a new population (return to Step 2).

6. **Conclusion**

   Steps 2 through 5 are repeated in sequence until the optimization has converged or the stopping criteria have been met. When either of these things occurs, the optimization concludes.

MOGA Steps to Generate a New Population

The process MOGA uses to generate a new population has two main steps: **Cross-over** and **Mutation**.

1. **Cross-over**

   Cross-over combines (mates) two chromosomes (parents) to produce a new chromosome (offspring). The idea behind cross-over is that the new chromosome can be better than both of the parents if it takes the best characteristics from each of the parents. Cross-over occurs during evolution according to a user-definable cross-over probability.

- **Cross-over for Continuous Parameters**

  A cross-over operator that linearly combines two parent chromosome vectors to produce two new offspring according to the following equations:

$$\text{Offspring1}=\mathbf{a}*\text{Parent1}+(1-\mathbf{a})*\text{Parent2}$$
$$\text{Offspring2}=(1-\mathbf{a})*\text{Parent1}+\mathbf{a}*\text{Parent2}$$

Consider the following two parents (each consisting of four floating genes), which have been selected for cross-over:

$$\text{Parent 1: } (0.3)(1.4)(0.2)(7.4)$$
$$\text{Parent 2: } (0.5)(4.5)(0.1)(5.6)$$

If a = 0.7, the following two offspring would be produced:

$$\text{Offspring1: } (0.36)(2.33)(0.17)(6.86)$$
$$\text{Offspring2: } (0.402)(2.981)(0.149)(6.842)$$

- **Cross-over for Discrete Parameters and Continuous Parameters with Manufacturable Values**

Each discrete parameter or continuous parameter with manufacturable values is represented by a binary chain corresponding to the number of levels. For example, a parameter with two values (levels) is encoded to one bit, a parameter with seven values is encoded to three bits, and an -bits chain represents a parameter with values.

The concatenation of these chains forms the chromosome, which crosses over with another chromosome.

Three different kinds of cross-over are available:

- One-Point

A one-point cross-over operator that randomly selects a cross-over point within a chromosome then interchanges the two parent chromosomes at this point to produce two new offspring.

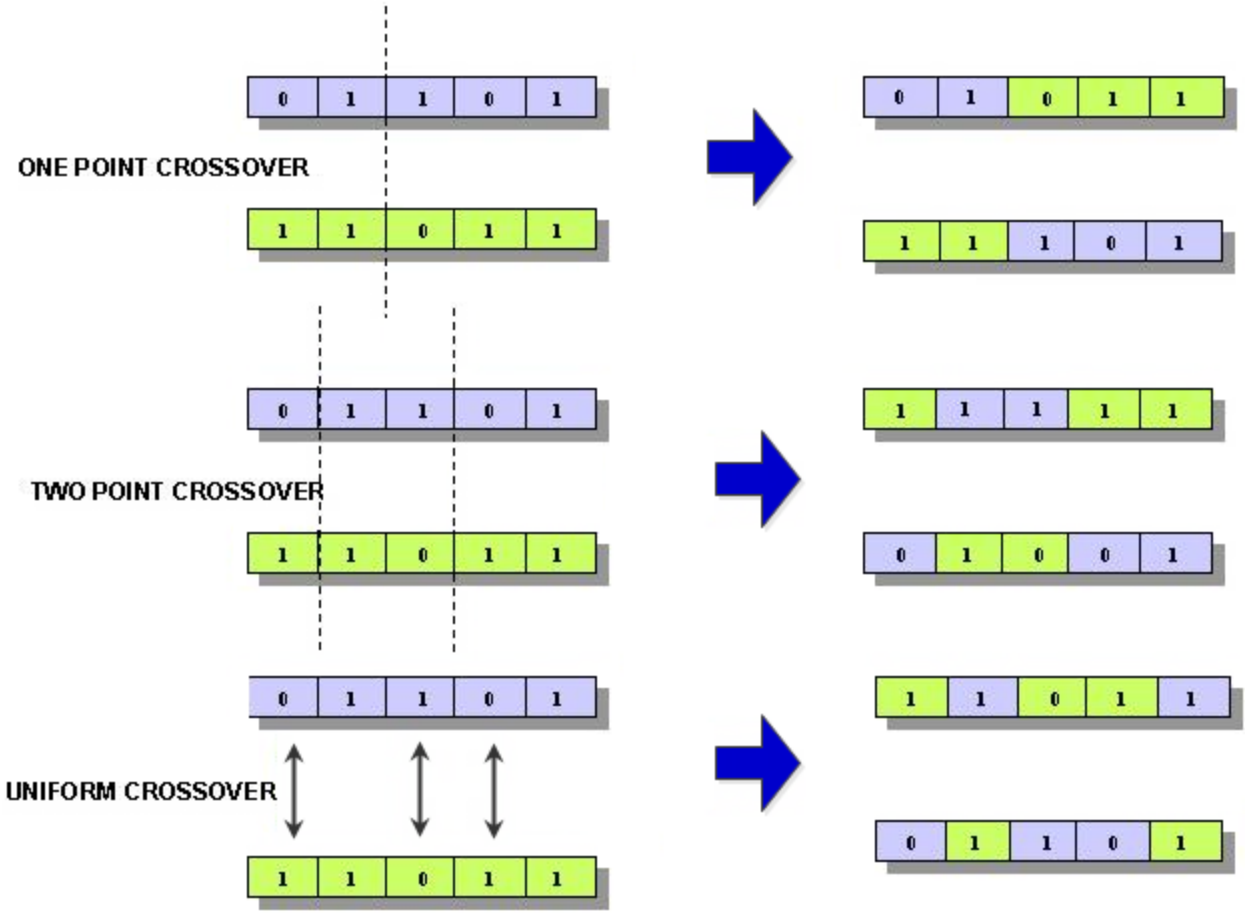
- Two-Point

A two-point cross-over operator randomly selects two cross-over points within a chromosome then interchanges the two parent chromosomes between these points to produce two new offspring.

- Uniform

A uniform cross-over operator decides (with some probability, which is known as the "mixing ratio") which parent contributes each of the gene values in the offspring chromosomes. This allows the parent chromosomes to be mixed at the gene level rather than the segment level (as

with one and two-point cross-over). For some problems, this additional flexibility outweighs the disadvantage of destroying building blocks.



2. **Mutation**

   Mutation alters one or more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With these new gene values, the genetic algorithm might be able to arrive at a better solution than was previously possible. Mutation is an important part of the genetic search, as it helps to prevent the population from stagnating at any local optima. Mutation occurs during evolution according to a user-defined mutation probability.

   - **Mutation for Continuous Parameters**

   For continuous parameters, a polynomial mutation operator is applied to implement mutation.

$$C = P + (\text{UpperBound} - \text{LowerBound})\delta$$

   where C is the child, P is the parent, and δ is a small variation calculated from a polynomial distribution.

- **Mutation for Discrete Parameters and Continuous Parameters with Manufacturable Values**

  For discrete parameters or continuous parameters with manufacturable values, a mutation operator simply inverts the value of the chosen gene (0 goes to 1 and 1 goes to 0) with a probability of 0.5. This mutation operator can only be used for binary genes. The concatenation of these chains forms the chromosome, which crosses over with another chromosome.

Related Topics

Setup Multi-Objective Genetic Algorithm (MOGA) Optimizer

Optimization Variables in Design Space

Cost Function

Convergence Criteria in MOGA-Based Multi-Objective Optimization

Convergence criteria are the conditions that indicate when the optimization has converged. In the Multi-Objective Genetic Algorithm (MOGA)-based multi-objective optimization methods, the following convergence criteria are available:

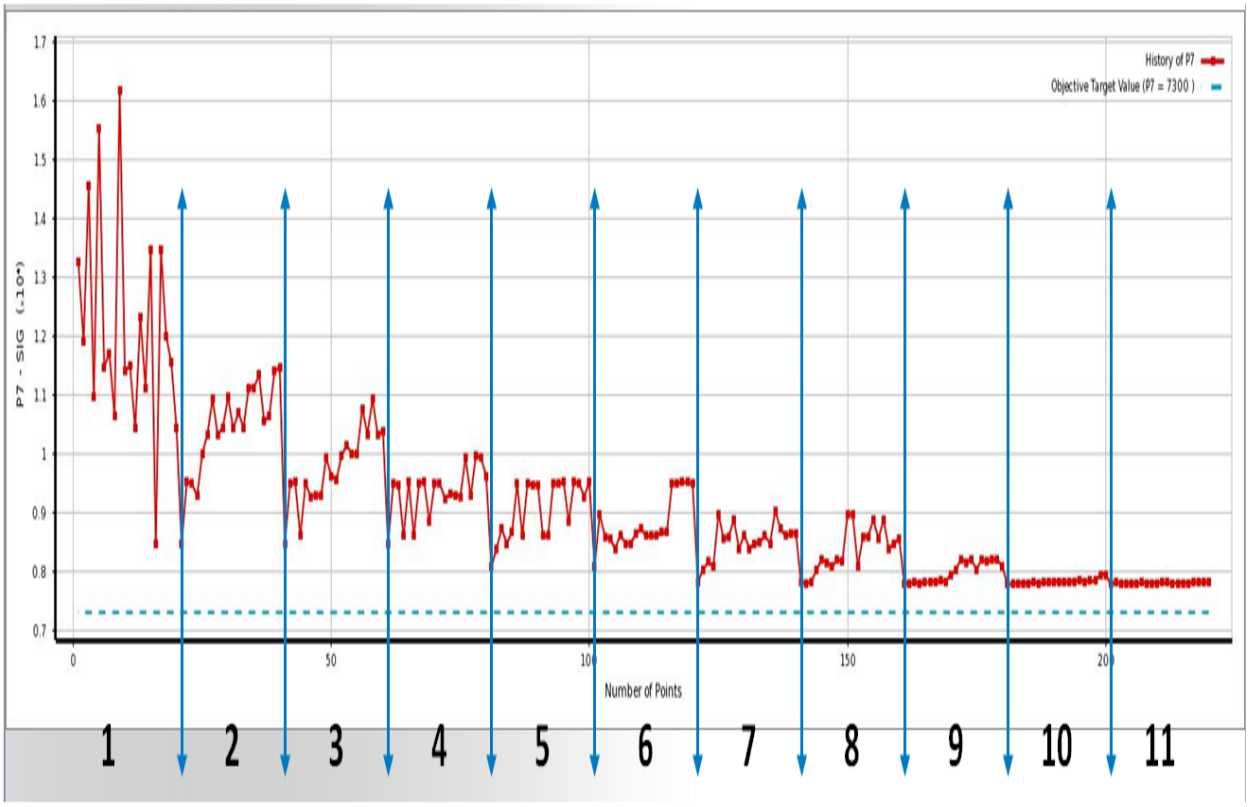
Maximum Allowable Pareto Percentage

The **Maximum Allowable Pareto Percentage** criterion looks for a percentage that represents a specified ratio of Pareto points per Number of Samples Per Iteration. When this percentage is reached, the optimization is converged.

Convergence Stability Percentage

The **Convergence Stability Percentage** criterion looks for population stability, based on mean and standard deviation of the output parameters. When a population is stable with regards to the previous one, the optimization is converged. The criterion functions in the following sequence:

- **Population 1**: When the optimization is run, the first population is not taken into account. Because this population was not generated by the MOGA algorithm, it is not used as a range reference for the output range (for scaling values).
- **Population 2**: The second population is used to set the range reference. The minimum, maximum, range, mean, and standard deviation are calculated for this population.
- **Populations 3 – 11**: Starting from the third population, the minimum and maximum output values are used in the next steps to scale the values (on a scale of 0 to 100). The mean variations and standard deviation variations are checked. If both of these are smaller than the value for the Convergence Stability Percentage property, the algorithm is converged.



At each iteration and for each active output, convergence occurs if:

$$\frac{|Mean_i - Mean_{i-1}|}{Max - Min} < \frac{S}{100}$$

and

$$\frac{|StdDev_i - StdDev_{i-1}|}{Max - Min} < \frac{S}{100}$$

Where:

S=Stability Percentage

$Mean_i$= Mean of the *i*-th Population

$StdDev_i$ = Standard Deviation of the *i*-th Population

Max = Maximum Output Value calculated on the first generated population of MOGA

Min = Minimum Output Value calculated on the first generated population of MOGA

Related Topics

Convergence Rate % and Initial Finite Difference Delta % in NLPQ and MISQP

Typically, the use of Nonlinear Programming by Quadratic Lagrangian (NLPQL) or Mixed-Integer Sequential Quadratic Programming (MISQP) optimizers is suggested for continuous problems when there is only one objective function. The problem might or might not be constrained and must be analytic. This means that the problem must be defined only by continuous input parameters and that the objective functions and constraints should not exhibit sudden "jumps" in their domain.

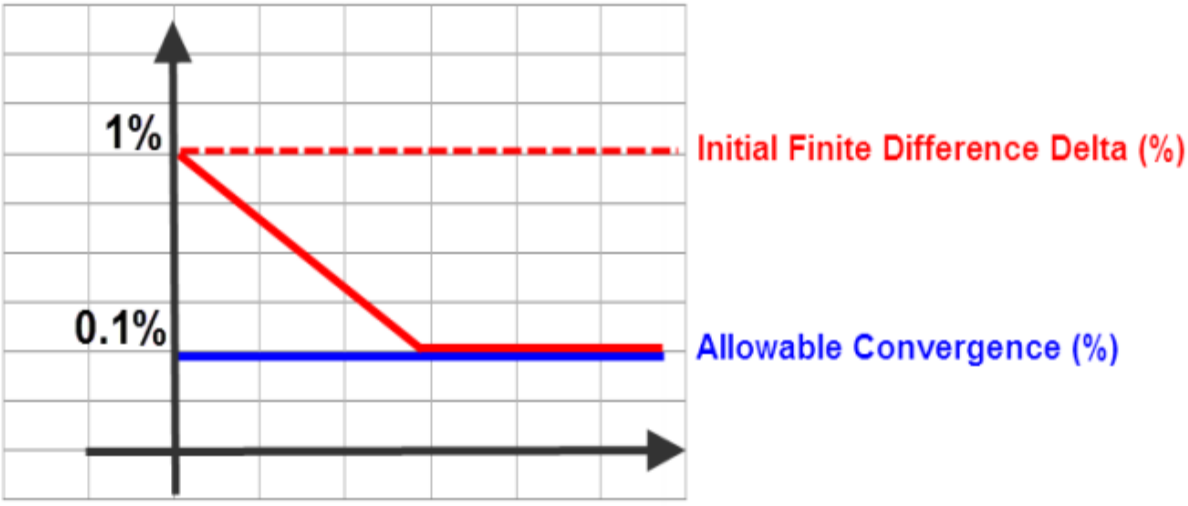
The main difference between these algorithms and Multi-Objective Genetic Algorithm (MOGA) is that MOGA is designed to work with multiple objectives and does not require full continuity of the output parameters. However, for continuous single objective problems, the use of NLPQL or MISQP gives greater accuracy of the solution as gradient information and line search methods are used in the optimization iterations. MOGA is a global optimizer designed to avoid local optima traps, while NLPQL and MISQP are local optimizers designed for accuracy.

For NLPQL and MISQP, the default convergence rate, which is specified by the Allowable Convergence (%) property, is set to 0.1% for a Direct Optimization system. The maximum value for this property is 100%. This is computed based on the (normalized) Karush-Kuhn-Tucker (KKT) condition. This implies that the fastest convergence rate of the gradients or the functions (objective function and constraint) determine the termination of the algorithm.

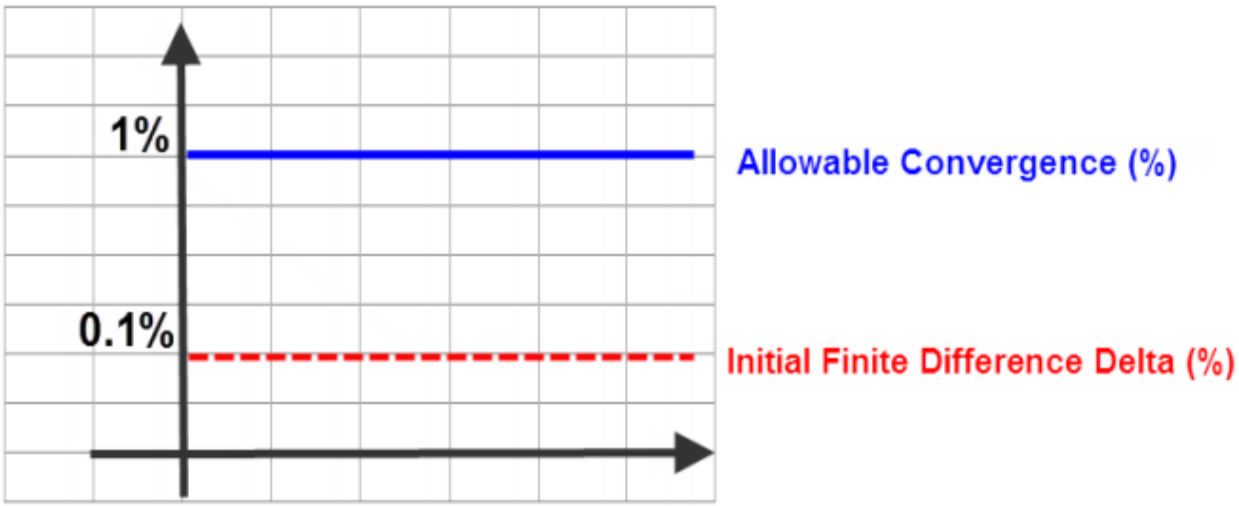
The default convergence rate is used in conjunction with the initial finite difference delta percentage value, which is specified by the Initial Finite Difference Delta (%) advanced property. This property defaults to 1% for a Direct Optimization system. You use this property to specify a percentage of the variation between design points to ensure that the Delta use in the calculation of finite differences is large enough to be seen over simulation noise. The specified percentage is defined as a relative gradient perturbation between design points.

The advantage of this approach is that for large problems, it is possible to get a near-optimal feasible solution quickly without being trapped into a series of iterations involving small solution steps near the optima. To work most effectively with NLPQL and MISQP, keep the following guidelines in mind:

- If the Initial Finite Difference Delta (%) is greater than the Allowable Convergence (%), the relative gradient perturbation gets iteratively smaller, until it matches the allowable convergence rate. At this point, the relative gradient value stays the same through the rest of the analysis.



- If the Initial Finite Difference Delta (%) is less than or equal to the Allowable Convergence (%), the current relative gradient step remains constant through the rest of the analysis.



- Both the Initial Finite Difference Delta (%) and Allowable Convergence (%) should be higher than the magnitude of the noise in your simulation.

When setting the values for these properties, you have the usual trade-offs between speed and accuracy. Smaller values result in more convergence iterations and a more accurate (but slower) solution, while larger values result in fewer convergence iterations and a less accurate (but faster) solution. At the same time, however, you must be aware of the amount of noise in your model. For the input variable variations to be visible in the output variables, both values must be greater than the magnitude of the simulation's noise.

In general, default values for Initial Finite Difference Delta (%) and Allowable Convergence (%) cover the majority of optimization problems. For example, if you know that the noise magnitude in

your direct optimization problem is 0.0001, then the default values (Allowable Convergence (%) = 0.001 and Initial Finite Difference Delta (%) = 0.01) are good.

When the defaults are not a good match for your problem, of course, you can adjust the values to better suit your model and your simulation needs. If you require a more numerically accurate solution, you can set the convergence rate to as low as 1.0E-10% and then set the Initial Finite Difference Delta (%) accordingly.

Related Topics

Mixed-Integer Sequential Quadratic Programming

Mixed-Integer Sequential Quadratic Programming (MISQP) is a mathematical optimization algorithm as developed by Oliver Exler, Thomas Lehmann and Klaus Schittkowski (NLPQL). This method solves Mixed-Integer Non-Linear Programming (MINLP) of the form:

Minimize:

$$f(x,y)$$

Subject to:

$$g_j(x,y)=0,\ j=1,\ldots,m_e,$$

$$g_j(x,y)\geq 0,\ j=m_e+1,\ldots,m$$

Where:

$$x\in\mathbb{R}^{n_c}, y\in\mathbb{N}^{n_i}$$

$$x_l\leq x\leq x_u$$

$$y_l\leq y\leq y_u$$

The symbols and denote the vectors of the continuous and integer variables, respectively. It is assumed that problem functions and are continuously differentiable subject to all . It is not assumed that integer variables can be relaxed. In other words, problem functions are evaluated only at integer points and never at any fractional values in between.

MISQP solves MINLP by a modified sequential quadratic programming (SQP) method. After linearizing constraints and constructing a quadratic approximation of the Lagrangian function, mixed-integer quadratic programs are successively generated and solved by an efficient branch-and-cut method. The algorithm is stabilized by a trust region method as originally proposed by Yuan for continuous programs. Second order corrections are retained. The Hessian of the

Lagrangian function is approximated by BFGS updates subject to the continuous and integer variables. MISQP is able to solve also non-convex nonlinear mixed-integer programs.

Related Topics

Setup Mixed Integer Quadratic Sequential Optimization

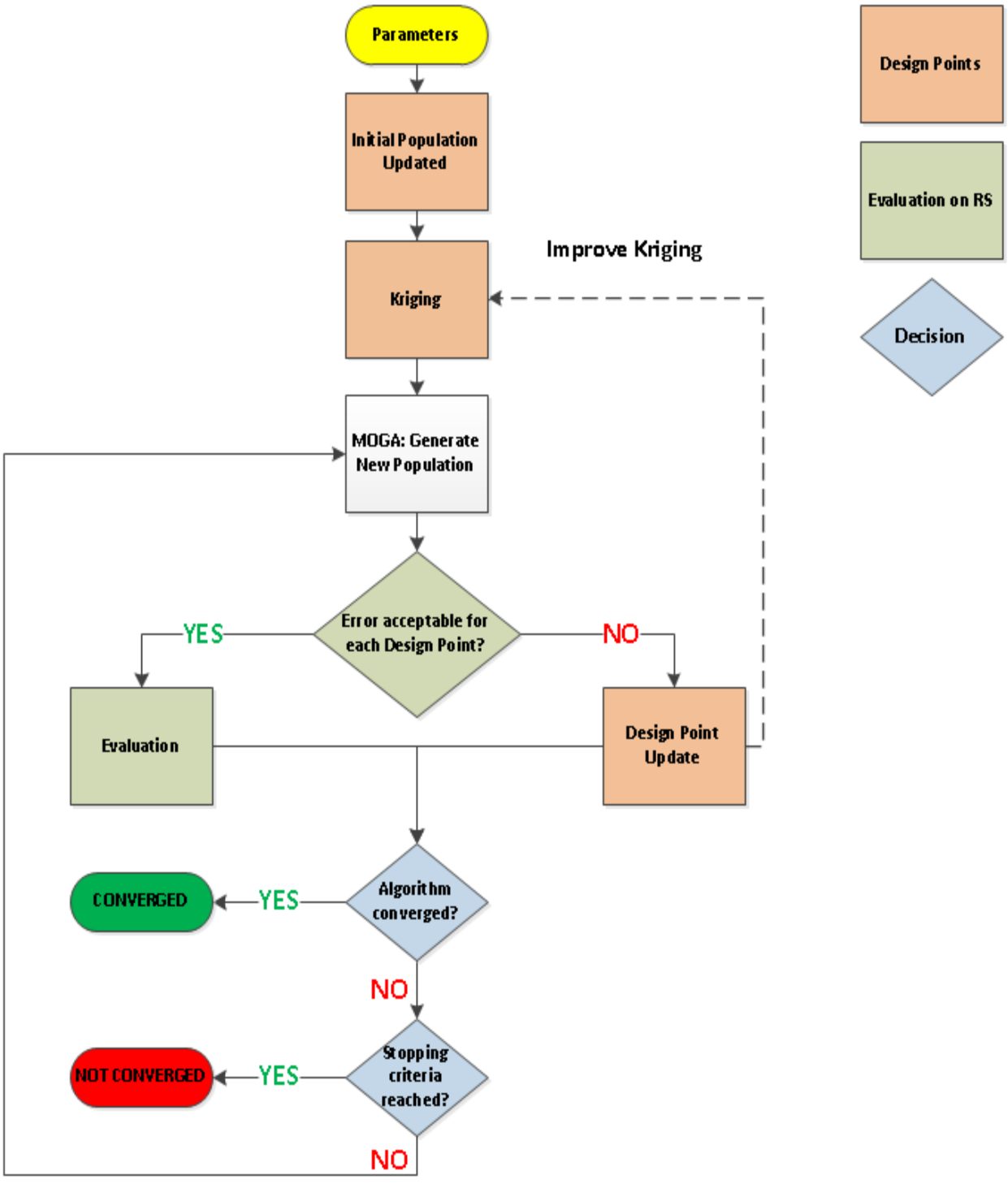
Adaptive Multiple Objective Optimization

Adaptive Multiple-Objective (AMO) is a mathematical optimization that combines a Kriging response surface and MOGA. It allows you to either generate a new sample set or use an existing set, providing a more refined approach than the Screening method. Except when necessary, the optimizer does not evaluate all design points. The general optimization approach is the same as MOGA, but a Kriging response surface is used. Part of the population is "simulated" by evaluations of the Kriging response surface. The Kriging error predictor reduces the number of evaluations used in finding the first Pareto front solutions.

AMO supports multiple objectives and multiple constraints. It is limited to continuous parameters, including those with manufacturable values. It is available only for a Direct Optimization system. When discrete parameters are used, MOGA is the more efficient optimization method. For more information, see Multi-Objective Genetic Algorithm (MOGA).

AMO Workflow

The workflow of AMO follows:



AMO Steps

1. **First Population of MOGA**

   The initial population is used to run MOGA.

2. **Kirging Generation**

   A Kriging response surface is created for each output, based on the first population and then improved during simulation with the addition of new design points.

3. **MOGA**

   MOGA is run, using the Kriging response surface as an evaluator. After the first iteration, each population is run when it reaches the number of samples defined by the Number of Samples Per Iteration property.

4. **Evaluate the Population**

5. **Error Check**

   The Kriging error predictor is checked for each point.

   - **Yes: Error Acceptable**

     Each point is validated for error. If the error for a given point is acceptable, the approximated point is included in the next population to be run through MOGA (return to Step 3).

   - **No: Error Not Acceptable**

     If the error is not acceptable, the points are promoted as design points. The new design points are used to improve the Kriging response surface (return to Step 2) and are included in the next population to be run through MOGA (return to Step 3)

6. **Convergence Validation**

   The optimization is validated for convergence.

   - **Yes: Optimization Converged**

   MOGA converges when the maximum allowable Pareto percentage has been reached. When this happens, the process is stopped.

   - **No: Optimization Not Converged**

     If the optimization is not converged, the process continues to the next step.

7. **Stopping Criteria Validation**

   If the optimization has not converged, it is validated for fulfillment of the stopping criteria.

   - **Yes: Stopping Criteria Met**

   When the maximum number of iterations has been reached, the process is stopped without having reached convergence..

   - **No: Stopping Criteria Not Met**

   If the stopping criteria have not been met, the MOGA algorithm is run again (return to Step 3).

8. **Conclusion**

   Steps 2 through 7 are repeated in sequence until the optimization has converged or the stopping criteria have been met. When either of these things occurs, the optimization concludes.

Related Topics

Setup Multi-Objective Genetic Algorithm (MOGA) Optimizer

Optimization Variables in Design Space

Cost Function

Adaptive Single Objective Optimization

Adaptive Single-Objective (ASO) is a mathematical optimization method that combines an OSF (Optimal Space-Filling) DOE, a Kriging response surface, and MISQP. It is a gradient-based algorithm based on a response surface, which provides a refined, global, optimized result.

ASO supports a single objective and multiple constraints. It is available for continuous parameters, including those with manufacturable values. It does not support the use of parameter relationships in the optimization domain and is available only for a Direct Optimization system.

Like MISQP, ASO solves constrained nonlinear programming problems of the form:

Minimize:

$$f = f(\{x\})$$

Subject to:

$$g_k(\{x\}) \leq 0, \forall k = 1,2,\ldots,K$$

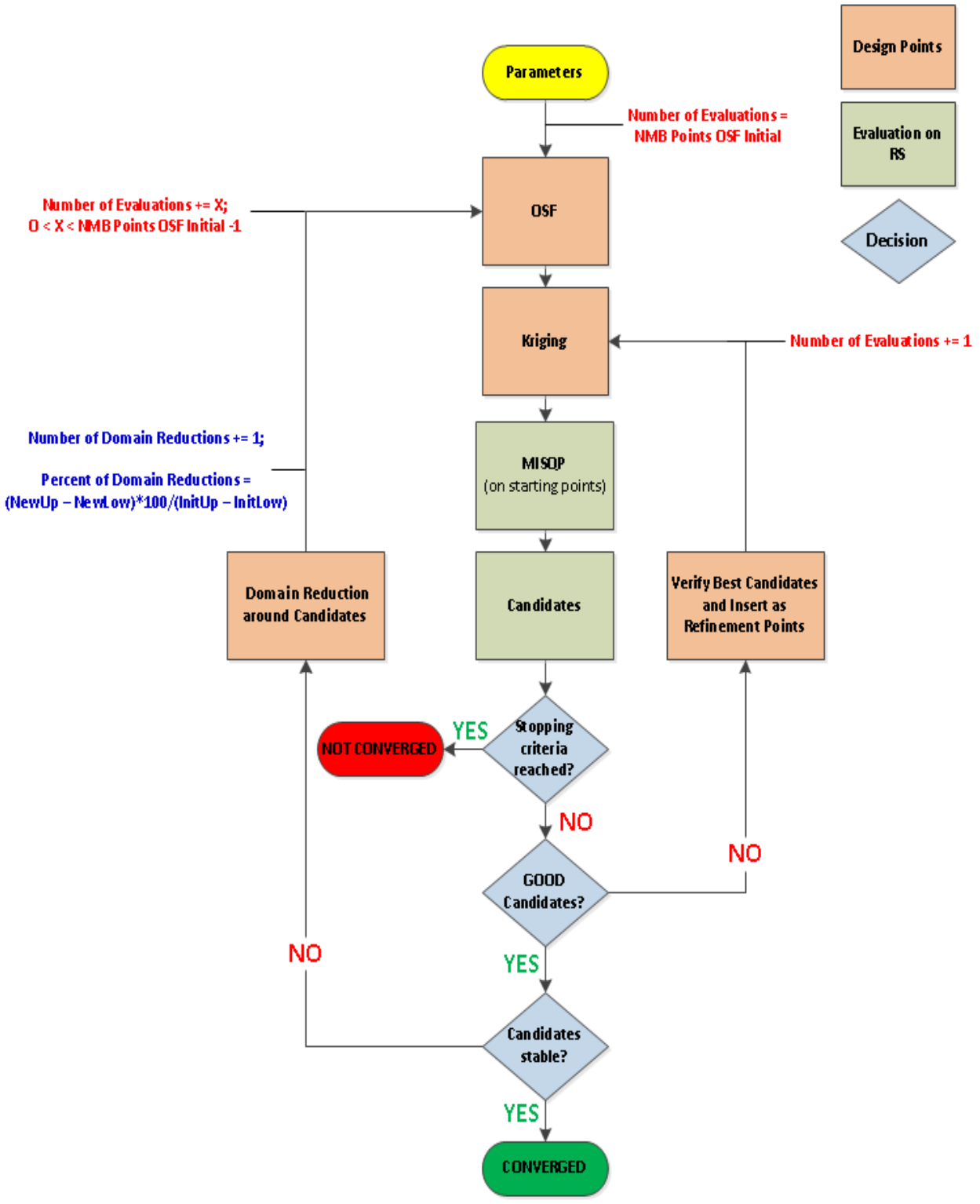
$$h_l(\{x\}) = 0, \forall I = 1,2\ldots,L$$

Where:

$$\{x_L\} \leq \{x\} \leq \{x_U\}$$

ASO Workflow

The workflow of ASO follows:



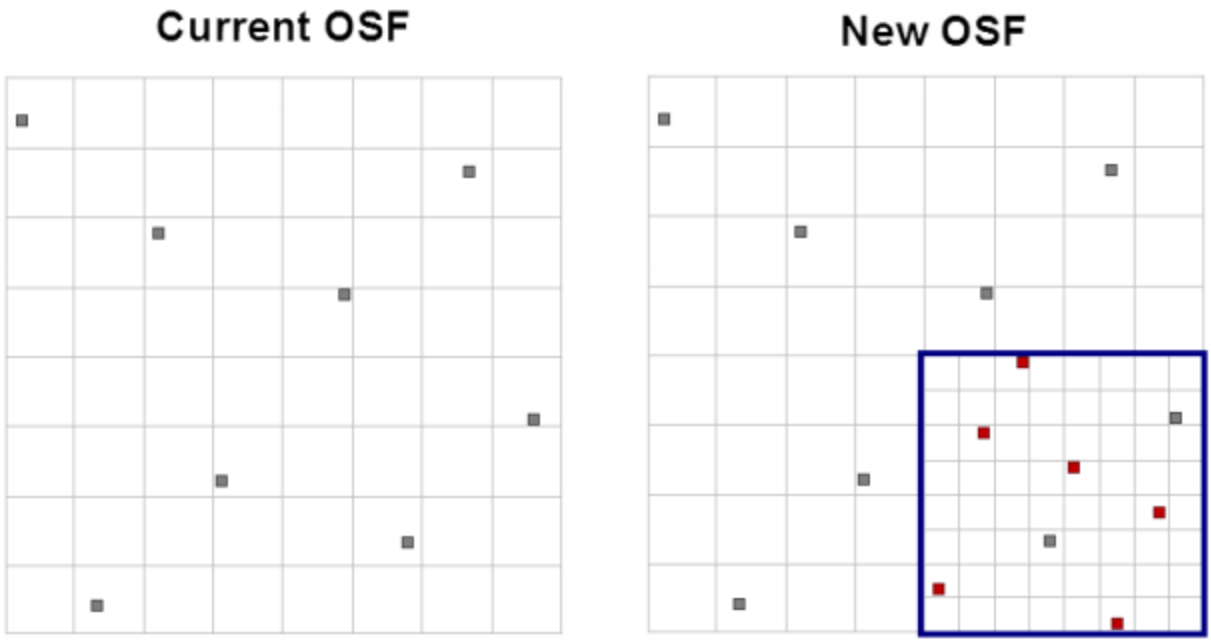
AMO Steps

1. **OSF Sampling**

   OSF (Optimal Space-Filling Design) is used for the Kriging construction. In the original OSF, the number of samples equals the number of divisions per axis and there is one sample in each division.

   When a new OSF is generated after a domain reduction, the reduced OSF has the same number of divisions as the original and keeps the existing design points within the new bounds. New design points are added until there is a point in each division of the reduced domain.

   In the following example, the original domain has eight divisions per axis and contains eight design points. The reduced domain also has eight divisions per axis and includes two of the original design points. To have a design point in each division, six new design points need to be added.



2. **Kirging Generation**

   A response surface is created for each output, based on the current OSF and consequently on the current domain bounds.

   A Kriging response surface is created for each output, based on the first population and then improved during simulation with the addition of new design points.

   For more information on Kriging algorithms, see Kriging Algorithms.

3. **MISQP**

   MISQP is run on the current Kriging response surface to find potential candidates. A few MISQP processes are run at the same time, beginning with different starting points, and consequently, giving different candidates.

4. **Candidate Point Validation**

   All the obtained candidates are either validated or not, based on the Kriging error predictor. The candidate point is checked to see if further refinement of the Kriging surface changes the selection of this point. A candidate is considered as acceptable if there aren't any points, according to this error prediction, that call it into question. If the quality of the candidate is not called into question, the domain bounds are reduced. Otherwise, the candidate is calculated as a verification point.

   - **Refinement Point Creation** (If the selection is ***not*** to be changed)

     When a new verification point is calculated, it is inserted in the current Kriging response surface as a refinement point and the MISQP process is restarted.

   - **Domain Reduction**(If the selection is to be changed)

     When candidates are validated, new domain bounds must be calculated. If all of the candidates are in the same zone, the bounds are reduced, centered on the candidates. Otherwise, the bounds are reduced as an inclusive box of all candidates. At each domain reduction, a new OSF is generated (conserving design points between the new bounds) and a new Kriging response surface is generated based on this new OSF.

5. **Convergence and Stop Criteria**

   The optimization is considered to be converged when the candidates found are stable. This occurs when all of the MISQP processes run on the response surface converge to the same verified candidate point. However, there are four stop criteria that can stop the algorithm: Maximum Number of Evaluations, Maximum Number of Domain Reductions, Percentage of Domain Reductions, and Convergence Tolerance.

Related Topics

Optimization Variables in Design Space

Cost Function

Optimization Variables and the Design Space

Once the optimization variables are specified, the optimizer handles each of them as an *n*-dimensional vector *x*. Any point in the design space corresponds to a particular *x*-vector and to a design instance. Each design instance may be evaluated via FEA and assigned a cost value;

therefore, the cost function is defined over the design space (cost(*x*): $R^n \rightarrow R$, where *n* is the number of optimization variables.

In practice, a solution of the minimization problem is sought only on a bounded subset of the Rn space. This subset is called the feasible domain and is defined via linear constraints.

Setting Up an Optimization Analysis

Optimization allows you to vary predefined variables in the nominal design to search for the solution that best satisfies a set of user defined goals or cost functions. Optimetrics modifies the variable values until the minimum is reached with acceptable accuracy.

Note	• You can define more than one optimization analysis setup per design. • You can create an Optimization setup before defining variables but all variables must be defined before you start the Optimization analysis. • Once you have created an optimization analysis setup, you can copy and paste it, and then make changes to the copy, rather than redoing the whole process for minor changes.

To provide a broad range of capability, Optimetrics incorporates the following types of numerical optimizers:

- Sequential Nonlinear Programming (Gradient) (SNLP)
- Sequential Mixed Integer Nonlinear Programming (Gradient and Discrete) (SMINLP)
- Quasi Newton (Gradient)
- Pattern Search (Search-based)
- Genetic Algorithm (Random search)
- MATLAB
- Screening (Search based)
- Multi-Objective Genetic Algorithm
- Non-linear Programming by Quadratic Lagrangian (Gradient)
- Mixed-Integer Sequential Quadratic Programming (Gradient and Discrete)
- Adaptive Multiple Objective (Gradient)
- Adaptive Single Objective (Gradient)

Click on the links above to view the setup procedure for each optimizer. Options for the analysis are listed in the table.

The following ***optional*** optimization solution setup options can also be used:

- Modify the starting variable value.
- Modify the minimum and maximum values of variables that will be optimized.
- Exclude variables from optimization.
- Modify the values of fixed variables that are not being optimized.
- Set the minimum and maximum step size between solved design variations (For the Quasi Newton and Patterns Search optimizers, Variables tab).
- Set the minimum and maximum focus size. (For the SNLP and SMINLP optimizers, Variables tab).
- Set Linear constraints.
- Request that Optimetrics solve a parametric sweep before an optimization analysis.
- Request that Optimetrics solve a parametric sweep during an optimization analysis.
- Automatically update optimized variables to the optimal values during an optimization or after an optimization analysis is completed.

- Change the norm used for the cost function calculation (Advanced Option)
- Set **HPC and Analysis Options**, which allows you to select or create an analysis configuration.

Note	Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

Related Topics

Optimization Overview

Choosing an Optimizer

Optimization Setup for the Quasi Newton(Gradient) Optimizer

Following is the procedure for setting up an optimization analysis using the Quasi Newton (gradient) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

1. Set up the variables you want to optimize in the **Design Properties** dialog box.
2. On the **Maxwell3D or Maxwell2D** menu, point to **Optimetrics Analysis**, and then click **Add Optimization**  .

   The **Setup Optimization** dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **Quasi Newton(Gradient)** from the **Optimizer** pull-down list. Selecting Quasi Newton enables the **Acceptable Cost** and **Noise** fields.
4. Type the maximum number of iterations you want Optimetrics to perform during the optimization analysis in the **Max. No. of Iterations** text box.
5. Under **Cost Function**, add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.
6. Type the value of the cost function at which the optimization process should stop in the **Acceptable Cost** text box.
7. Type the cost function noise in the **Noise** text box.
8. If you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

   The **Cost Function Norm Type** pull-down list appears.

   - Select **L1**, **L2**, or **Maximum**.

   A norm is a function that assigns a positive value to the cost function.

For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.

9. Optionally, click the button for setting **HPC and Analysis Options**, which allows you to select or create an analysis configuration.
10. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization, and the **Min/Max Step Size** for the analysis.
    - You may also override the variable starting values by clicking the **Override** check box and entering the desired value in the **Starting Value** field.
    - Optionally, modify the values of fixed variables that are not being optimized.
    - Optionally, set Linear constraints.
    - Select the **View all columns** check box to see all columns, including hidden columns.
11. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

    Enabling the **Update design parameters' value after optimization** check box will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.
12. Under the **Options** tab, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note	Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

    You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation.This will provide some speed improvement in the overall optimization process.

Optimization Setup for the Pattern Search (Search-based) Optimizer

Following is the procedure for setting up an optimization analysis using the Pattern Search (Search-based) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

1. Set up the variables you want to optimize in the **Design Properties** dialog box.
2. On the **Maxwell3D or Maxwell2D** menu, point to **Optimetrics Analysis**, and then click **Add Optimization**  .

The **Setup Optimization** dialog box appears.

3. Under the **Goal**s tab, select the optimizer by selecting **Pattern Search(Search-based)** from the **Optimizer** pull-down list. Selecting Pattern Search enables the **Acceptable Cost** and **Noise** fields.
4. Type the maximum number of iterations you want Optimetrics to perform during the optimization analysis in the **Max. No. of Iterations** text box.
5. Under **Cost Function**, add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.
6. Type the value of the cost function at which the optimization process should stop in the **Acceptable Cost** text box.
7. Type the cost function noise in the **Noise** text box.
8. If you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

   The **Cost Function Norm Type** pull-down list appears.

   - Select **L1**, **L2**, or **Maximum**.

   A norm is a function that assigns a positive value to the cost function.

   For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

   The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.

9. Optionally, click the button for setting **HPC and Analysis Options**, which allows you to select or create an analysis configuration.
10. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization, and the **Min/Max Step Size** for the analysis.
    - You may also override the variable starting values by clicking the **Override** checkbox and entering the desired value in the **Starting Value** field.
    - Optionally, modify the values of fixed variables that are not being optimized.
    - Optionally, set Linear constraints.
    - Select the **View all columns** check box to see all columns, including hidden columns.
11. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

    Enabling the **Update design parameters' value after optimization** check box will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.

12. Under the **Options** tab, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note	Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation.This will provide some speed improvement in the overall optimization process.

Optimization Setup for the Sequential Nonlinear Programming (Gradient) Optimizer

Following is the procedure for setting up an optimization analysis using the Sequential Nonlinear Programming (Gradient) Optimizer or SNLP Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

1. Set up the variables you want to optimize in the **Design Properties** dialog box.
2. On the **Maxwell3D or Maxwell2D** menu, point to **Optimetrics Analysis**, and then click **Add Optimization**  .

   The **Setup Optimization** dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **Sequential Nonlinear Programming(Gradient)** from the **Optimizer** pull-down list.
4. Type the maximum number of iterations you want Optimetrics to perform during the optimization analysis in the **Max. No. of Iterations** text box.
5. Under **Cost Function**, add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.
6. If you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

   The **Cost Function Norm Type** pull-down list appears.

   - Select **L1**, **L2**, or **Maximum**.

   A norm is a function that assigns a positive value to the cost function.

   For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.

7. Optionally, click the button for setting **HPC and Analysis Options**, which allows you to select or create an analysis configuration.
8. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization, and the **Min/Max Focus** for the analysis.
   - You may also override the variable starting values by clicking the **Override** checkbox and entering the desired value in the **Starting Value** field.
   - Optionally, modify the values of fixed variables that are not being optimized.
   - Optionally, set Linear constraints.
   - Select the **View all columns** check box to see all columns, including hidden columns.
9. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

   Enabling the **Update design parameters' value after optimization** check box will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.
10. Under the **Options** tab, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note	Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

    You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation.This will provide some speed improvement in the overall optimization process.

Optimization Setup for the Sequential Mixed Integer Nonlinear Programming (Gradient and Discrete) Optimizer

Following is the procedure for setting up an optimization analysis using the Sequential Mixed Integer Nonlinear Programming (Gradient and Discrete) Optimizer or SMINLP. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

1. Set up the variables you want to optimize in the **Design Properties** dialog box.
2. On the **Maxwell3D or Maxwell2D** menu, point to **Optimetrics Analysis**, and then click

   Add Optimization .

   The **Setup Optimization** dialog box appears.

3. Under the **Goals** tab, select the optimizer by selecting **Sequential Mixed Integer Nonlinear Programming(Gradient and Discrete)** from the **Optimizer** pull-down list.
4. Type the maximum number of iterations you want Optimetrics to perform during the optimization analysis in the **Max. No. of Iterations** text box.
5. Under **Cost Function**, add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.
6. If you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

   The **Cost Function Norm Type** pull-down list appears.

   - Select **L1**, **L2**, or **Maximum**.

   A norm is a function that assigns a positive value to the cost function.

   For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

   The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.
7. Optionally, click the button for setting **HPC and Analysis Options**, which allows you to select or create an analysis configuration.
8. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization, and the **Min/Max Focus** for the analysis.
   - You may also override the variable starting values by clicking the **Override** checkbox and entering the desired value in the **Starting Value** field.
   - Optionally, modify the values of fixed variables that are not being optimized.
   - Optionally, set Linear constraints.
   - Select the **View all columns** check box to see all columns, including hidden columns.
9. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

   Enabling the **Update design parameters' value after optimization** check box will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.
10. Under the **Options** tab, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note	Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or

source excitation.This will provide some speed improvement in the overall optimization process.

Optimization Setup for the Genetic Algorithm (Random search) Optimizer

Following is the procedure for setting up an optimization analysis using the Genetic Algorithm (Random search) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

1. Set up the variables you want to optimize in the **Design Properties** dialog box.
2. On the **Maxwell3D or Maxwell2D** menu, point to **Optimetrics Analysis**, and then click **Add Optimization**  .

   The **Setup Optimization** dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **Genetic Algorithm(Random search)** from the **Optimizer** pull-down list.
4. Click the **Setup...** button to modify the Advanced Genetic Algorithm Optimizer Options.
5. Under **Cost Function**, add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.
6. If you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

   The **Cost Function Norm Type** pull-down list appears.

   - Select **L1**, **L2**, or **Maximum**.

   A norm is a function that assigns a positive value to the cost function.

   For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

   The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.
7. Optionally, click the button for setting **HPC and Analysis Options**, which allows you to select or create an analysis configuration.
8. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization, and the **Min/Max Focus** for the analysis.
   - You may also override the variable starting values by clicking the **Override** checkbox and entering the desired value in the **Starting Value** field.
   - Optionally, modify the values of fixed variables that are not being optimized.

- Optionally, set Linear constraints.
- Select the **View all columns** check box to see all columns, including hidden columns.

9. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

   Enabling the **Update design parameters' value after optimization** check box will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.

10. Under the **Options** tab, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note	Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

    You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation.This will provide some speed improvement in the overall optimization process.

Optimization Setup for the MATLAB Optimizer

Following is the procedure for setting up an optimization analysis using the MATLAB Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

1. Set up the variables you want to optimize in the **Design Properties** dialog box.
2. On the **Maxwell3D or Maxwell2D** menu, point to **Optimetrics Analysis**, and then click **Add Optimization**  .

   The **Setup Optimization** dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **MATLAB** from the **Optimizer** pull-down list. Selecting MATLAB enables the **Acceptable Cost** and **Noise** fields.
4. Click the **Setup...** button to modify the MATLAB Optimizer Options.
5. Type the maximum number of iterations you want Optimetrics to perform during the optimization analysis in the **Max. No. of Iterations** text box.
6. Under **Cost Function**, add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.
7. Type the value of the cost function at which the optimization process should stop in the **Acceptable Cost** text box.
8. Type the cost function noise in the **Noise** text box.
9. If you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

The **Cost Function Norm Type** pull-down list appears.

- Select **L1**, **L2**, or **Maximum**.

A norm is a function that assigns a positive value to the cost function.

For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.

10. Optionally, click the button for setting **HPC and Analysis Options**, which allows you to select or create an analysis configuration.
11. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization, and the **Min/Max Step Size** for the analysis.
    - You may also override the variable starting values by clicking the **Override** checkbox and entering the desired value in the **Starting Value** field.
    - Optionally, modify the values of fixed variables that are not being optimized.
    - Optionally, set Linear constraints.
    - Select the **View all columns** check box to see all columns, including hidden columns.
12. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

    Enabling the **Update design parameters' value after optimization** checkbox will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.
13. Under the **Options** tab, if you want to save the field solution data for every solved design variation in the optimization analysis, select **Save Fields And Mesh**.

Note	Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

    You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation. This will provide some speed improvement in the overall optimization process.

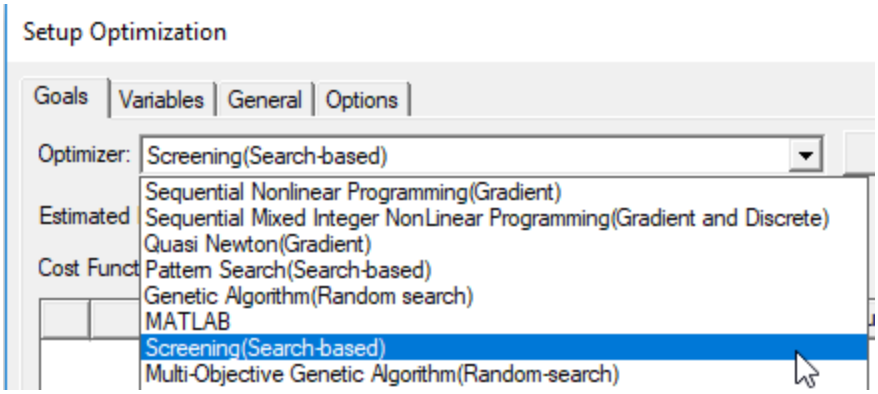
Setup Screening (Search Based) Optimizer

Following is the procedure for setting up an optimization analysis using the Screening Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

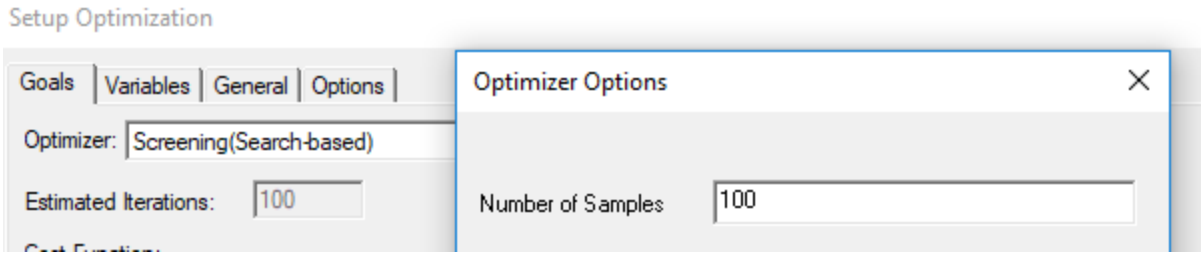
This is a non-iterative direct sampling method that uses a quasi-random number generator based on the Hammersley algorithm. You can start with Screening to locate the multiple tentative optima and then refine with NLPQL or MISQP to zoom in on the individual local maximum or minimum value. Usually Screening is used for preliminary design. Then, if you want to refine, these candidate points are used as starting points for gradient methods.

1. Set up the variables you want to optimize in the **Design Properties** dialog box. The variables must be swept in a Parametric setup.
2. Click **HFSS** or **Q3D Extractor** or **2D Extractor>Optimetrics Analysis**>**Add** **Optimization** .

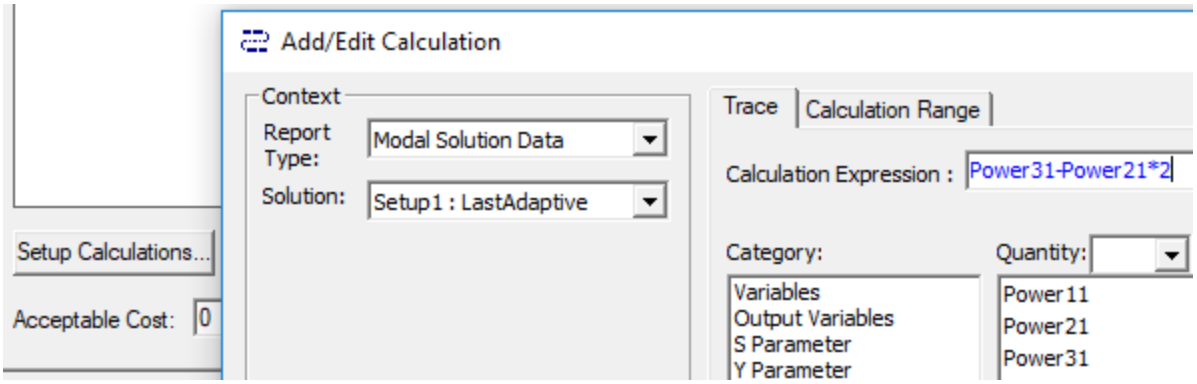
   The **Setup Optimization** dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **Screening (Search Based)** from the **Optimizer** drop-down menu.

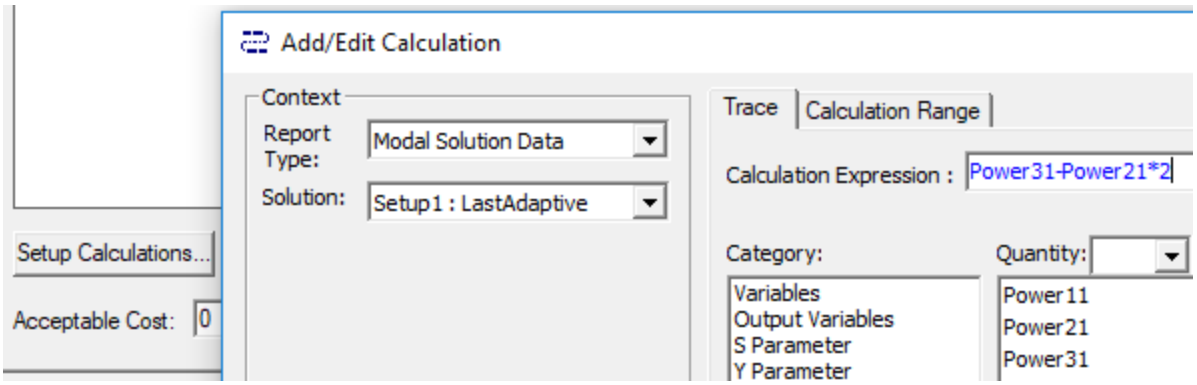


4. Optionally press the **Setup** button to open the **Optimizer Options window** to change the default number of samples from 100.



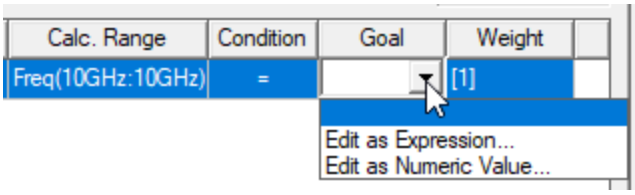
   The number of samples must be greater than the number of enabled input parameters. The number of enabled input parameters is also the minimum number of samples required to generate the Sensitivities chart. You can enter a minimum of 2 and a maximum of 10,000. The default is 100 for a Direct Optimization system.
5. Add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.



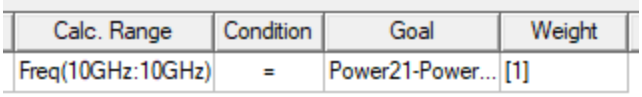


When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the **Add/EditCalculation** dialog.

6. In the Optimization setup, in the dropdown for the Goal column, select either Edit as Expression or Edit as Numeric Value...



This reopens the **Add/Edit Calculation** dialog box. If you are satisfied with the expression or value displayed, click **Done** to close the dialog box. This enters the expression/value to the **Goal** column.



7. In the **Optimization** setup, if you want to select a **Cost Function Norm Type**:

- Check the **Show Advanced Option** check box.

  The **Cost Function Norm Type** pull-down list appears.
- Select **L1**, **L2**, or **Maximum**.

  A norm is a function that assigns a positive value to the cost function.

  For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors, which means that it is always less than zero. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

  The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.

8. Optionally, set the Acceptable Cost and Cost Function Noise.
9. Optionally, click the button for setting HPC and Analysis Options, which allows you to select or create an analysis configuration.
10. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization.
    - You may also override the variable starting values by clicking the **Override** check box and entering the desired value in the **Starting Value** field.
    - Optionally, modify the values of fixed variables that are not being optimized.
    - Optionally, set Linear constraints.
    - Select the **View all columns** check box to see all columns, including hidden columns.
11. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

    Enabling the **Update design parameters' value after optimization** check box will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.
12. Under the **Options tab**, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note	Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

    You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation. This will provide some speed improvement in the overall optimization process.

Setup Multi-Objective Genetic Algorithm(Random Search) Optimizer

Following is the procedure for setting up an optimization analysis using the Multi-Objective Genetic Algorithm (Random Search) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

The Multi-Objective Genetic Algorithm (MOGA) is a hybrid variant of the popular NSGA-II (Non-dominated Sorted Genetic Algorithm-II) based on controlled elitism concepts. It supports all types of input parameters. The Pareto ranking scheme is done by a fast, non-dominated sorting method that is an order of magnitude faster than traditional Pareto ranking methods. The constraint handling uses the same non-dominance principle as the objectives. Therefore, penalty functions and Lagrange multipliers are not needed. This also ensures that the feasible solutions are always ranked higher than the infeasible solutions.

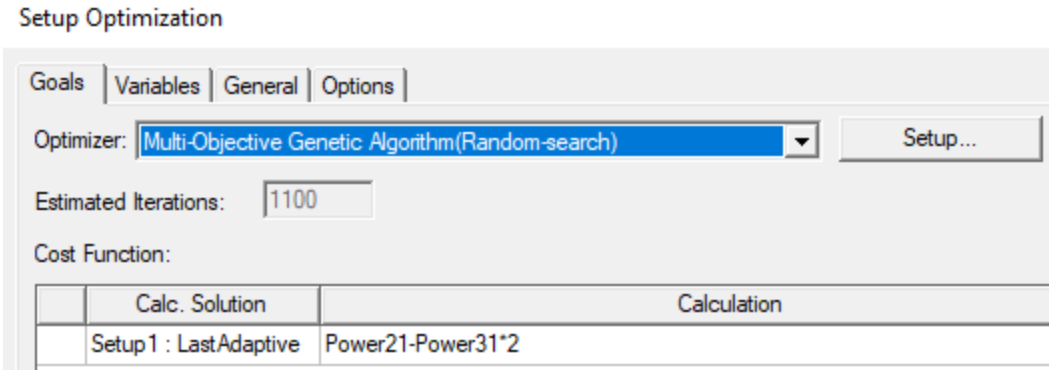
The first Pareto front solutions are archived in a separate sample set internally and are distinct from the evolving sample set. This ensures minimal disruption of Pareto front patterns already available from earlier iterations. You can control the selection pressure (and, consequently, the elitism of the process) to avoid premature convergence by altering the Maximum Allowable Pareto Percentage property.

1. Set up the variables you want to optimize in the **Design Properties** dialog box. The variables must be swept in a Parametric setup.
2. Click **HFSS** or **Q3D Extractor** or **2D Extractor>Optimetrics Analysis>Add**

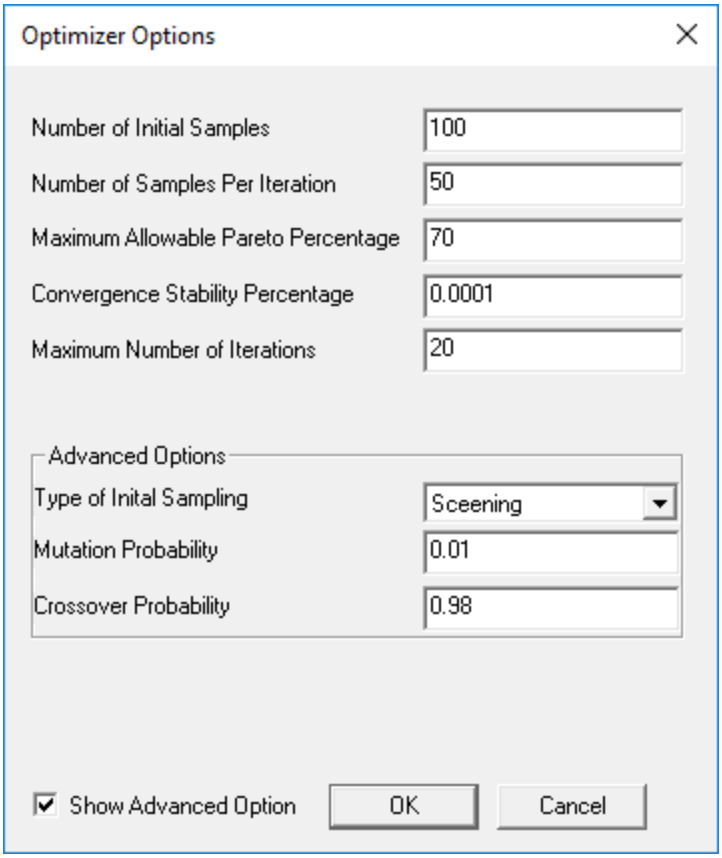
   

   Optimization .

   The **Setup Optimization** dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **Multi-Objective Genetic Algorithm (Random Search)** from the **Optimizer** drop-down menu.



4. Optionally press the **Setup** button to open the **Optimizer Options window**.



- **Number of Initial Samples**: Initial number of samples to use. This number must be greater than the number of enabled input parameters. The minimum recommended number of initial samples is 10 times the number of enabled input parameters. The larger the initial sample set, the better your chances of finding the input parameter space that contains the best solutions.

  The number of enabled input parameters is also the minimum number of samples required to generate the Sensitivities chart. You can enter a minimum of 2 and a maximum of 10000. The default is 100.

  If you switch from the Screening method to the MOGA method, MOGA generates a new sample set. For the sake of consistency, enter the same number of initial samples as you used for the Screening method.
- **Number of Samples Per Iteration**: Number of samples to iterate and update with each iteration. This number must be greater than the number of enabled input parameters but less than or equal to the number of initial samples. The default is for a Direct Optimization system.

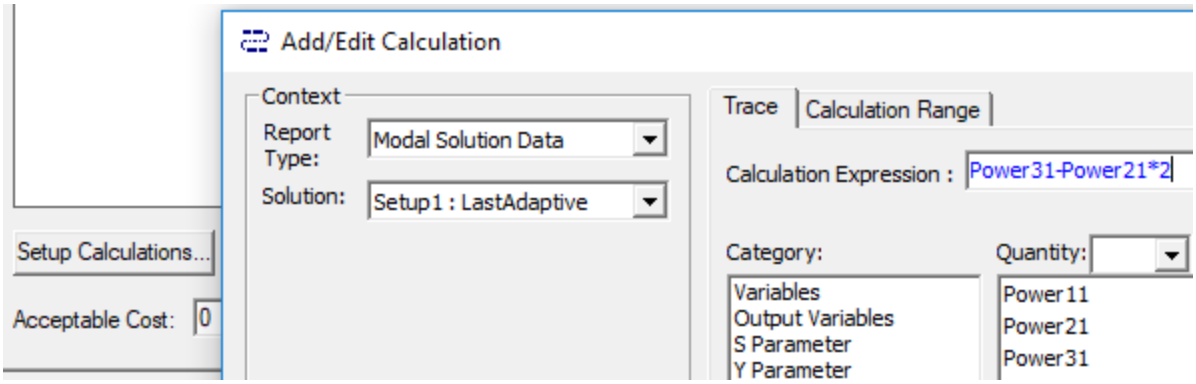
  You can enter a minimum of 2 and a maximum of 10000.
- **Maximum Allowable Pareto Percentage**: Convergence criterion. Percentage value that represents the ratio of the number of desired Pareto points to the number of samples per iteration. When this percentage is reached, the optimization is converged. For

example, a value of 70 with Number of Samples Per Iteration set to 200 would mean that the optimization should stop once the resulting front of the MOGA optimization contains at least 140 points. Of course, the optimization stops before that if the maximum number of iterations is reached.

- If the **Maximum Allowable Pareto Percentage** is too low (below 30), the process can converge prematurely. If the value is too high (above 80), the process can converge slowly. The value of this property depends on the number of parameters and the nature of the design space itself. The default is 70. Using a value between 55 and 75 works best for most problems. For more information, see Convergence Criteria in MOGA-Based Multi-Objective Optimization .
- **Convergence Stability Percentage**: Convergence criterion. Percentage value that represents the stability of the population based on its mean and standard deviation. This criterion allows you to minimize the number of iterations performed while still reaching the desired level of stability. When the specified percentage is reached, the optimization is converged. The default percentage is 2. To not take the convergence stability into account, set to 0. For more information, see Convergence Criteria in MOGA-Based Multi-Objective Optimization .
- **Maximum Number of Iterations**: Stop criterion. Maximum number of iterations that the algorithm is to execute. If this number is reached without the optimization having reached convergence, iterations stop. This also provides an idea of the maximum possible number of function evaluations that are needed for the full cycle, as well as the maximum possible time it can take to run the optimization. For example, the absolute maximum number of evaluations is given by:

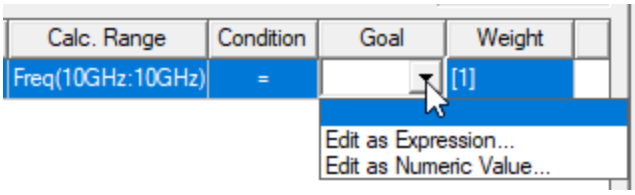
  Number of Initial Samples + Number of Samples Per Iteration * (Maximum Number of Iterations - 1)
- **Type of Initial Sampling**: Advanced option for generating different kinds of sampling. If you do not have any parameter relationships defined, set to Screening (default) or Optimal Space-Filling. If you have parameter relationships defined, the initial sampling must be performed by the constrained sampling algorithms because parameter relationships constrain the sampling. For such cases, this property is automatically set to Constrained Sampling.
- **Mutation Probability**: Advanced option for specifying the probability of applying a mutation on a design configuration. The value must be between 0 and 1. A larger value indicates a more random algorithm. If the value is 1, the algorithm becomes a pure random search. A low probability of mutation (<0.2) is recommended. The default is 0.01. For more information on mutation, see MOGA Steps to Generate a New Population.
- **Crossover Probability**: Advanced option for specifying the probability with which parent solutions are recombined to generate offspring solutions. The value must be between 0 and 1. A smaller value indicates a more stable population and a faster (but less accurate) solution. If the value is 0, the parents are copied directly to the new population. A high probability of crossover (>0.9) is recommended. The default is 0.98.

5. Add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.

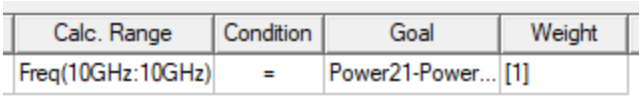


When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the **Add/EditCalculation** dialog.

6. In the Optimization setup, in the dropdown for the Goal column, select either **Edit as Expression...** or **Edit as Numeric Value...**.



This reopens the **Add/Edit Calculation** dialog box. If you are satisfied with the expression or value displayed, click **Done** to close the dialog box. This enters the expression/value to the **Goal** column.



7. In the **Optimization** setup, if you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

     The **Cost Function Norm Type** pull-down list appears.
   - Select **L1**, **L2**, or **Maximum**.

     A norm is a function that assigns a positive value to the cost function.

     For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors, which means that it is always less than zero. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

     The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.

8. Optionally, set the Acceptable Cost and Cost Function Noise.
9. Optionally, click the button for setting HPC and Analysis Options, which allows you to select or create an analysis configuration.
10. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization.
    - You may also override the variable starting values by clicking the **Override** check box and entering the desired value in the **Starting Value** field.
    - Optionally, modify the values of fixed variables that are not being optimized.
    - Optionally, set Linear constraints.
    - Select the **View all columns** check box to see all columns, including hidden columns.
11. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

    Enabling the **Update design parameters' value after optimization** check box will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.
12. Under the **Options tab**, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note	Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

    You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation. This will provide some speed improvement in the overall optimization process.

Setup Nonlinear Programming by Quadratic Lagrangian (Gradient) Optimizer

Following is the procedure for setting up an optimization analysis using the Nonlinear Programming by Quadratic Lagrangian (Gradient) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

The NLPQL (Nonlinear Programming by Quadratic Lagrangian) method can be used for Direct Optimization systems. It allows you to generate a new sample set to provide a more refined approach than the Screening method. Available for continuous input parameters only, NLPQL can handle only one output parameter goal. Other output parameters can be defined as constraints. For more information, see Convergence Rate % and Initial Finite Difference Delta % in NLPQL and MISQP and Nonlinear Programming by Quadratic Lagrangian (NLPQL).

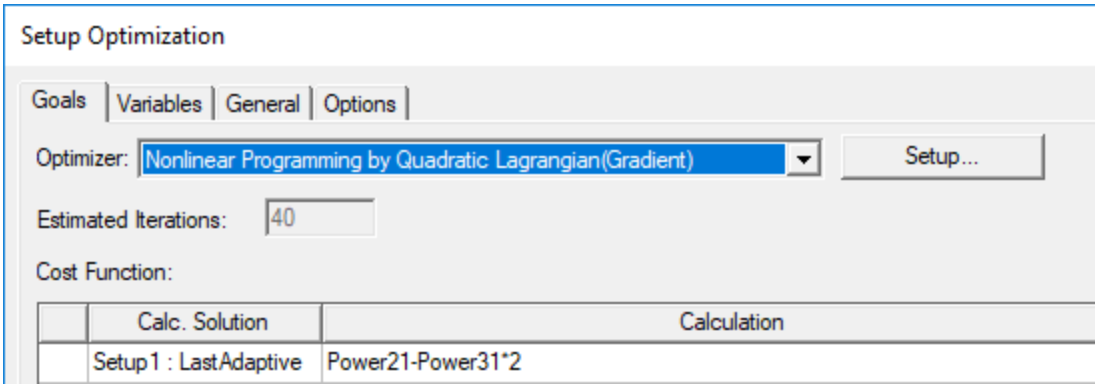
To generate samples and perform an NLPQL optimization:

1. Set up the variables you want to optimize in the **Design Properties** dialog box. The variables must be swept in a Parametric setup.
2. Click **HFSS** or **Q3D Extractor** or **2D Extractor>Optimetrics Analysis>Add**

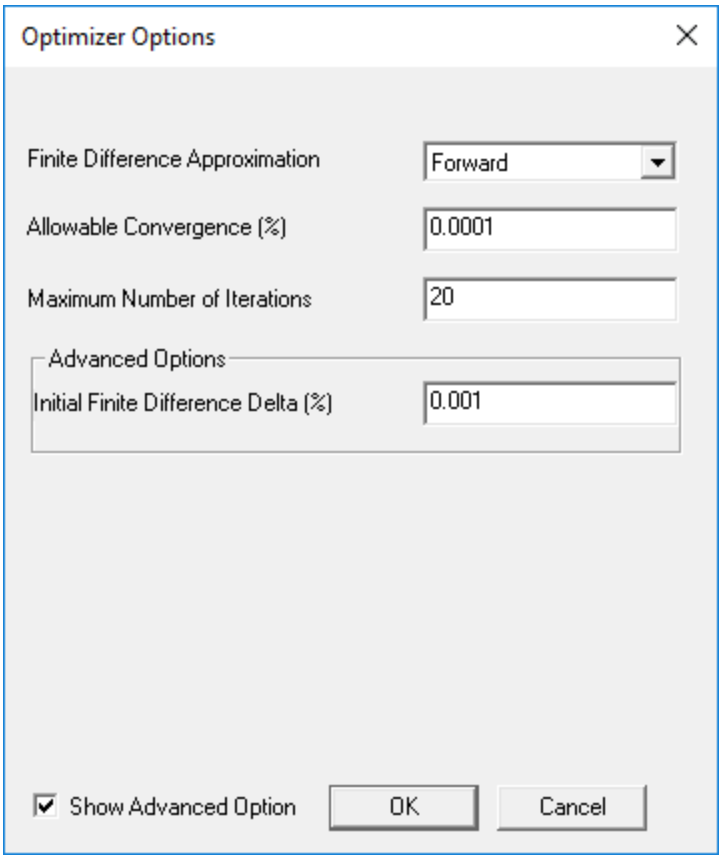
Optimization .

The **Setup Optimization** dialog box appears.

3. Under the **Goals** tab, select the optimizer by selecting **Nonlinear Programming by Quadratic Lagrangian (Gradient)** from the **Optimizer** drop-down menu.



4. Optionally press the **Setup** button to open the **Optimizer Options window**.



- **Finite Difference Approximation**: When analytical gradients are not available, NLPQL approximates them numerically. This property allows you to specify the method of approximating the gradient of the objective function. Choices are:

- **Central**: Increases the accuracy of the gradient calculations by sampling from both sides of the sample point but increases the number of design point evaluations by 50%. This method makes use of the initial point, as well as the forward point and rear point.
- **Forward**: Uses fewer design point evaluations but decreases the accuracy of the gradient calculations. This method makes use of only two design points, the initial point and forward point, to calculate the slope forward. This is the default method for new Direct Optimization systems.

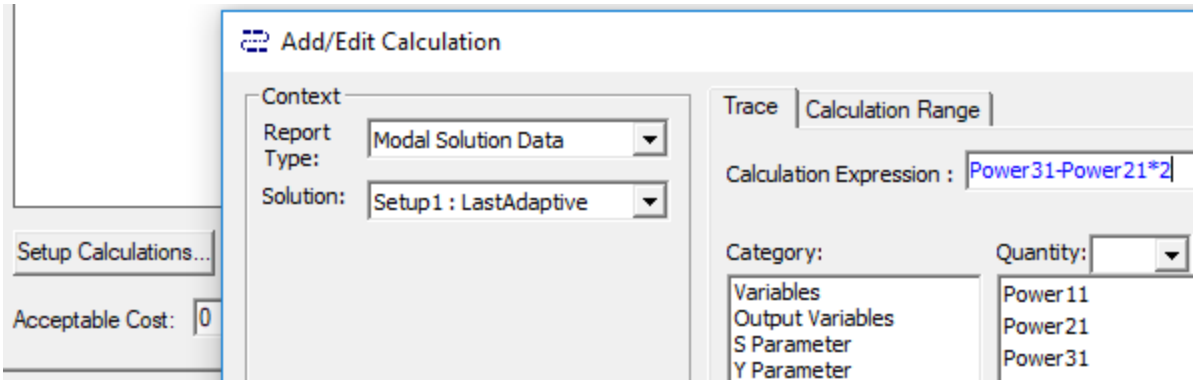
- **Maximum Number of Iterations**: Stop criterion. Maximum number of iterations that the algorithm is to execute. If convergence happens before this number is reached, the iterations stop. This also provides an idea of the maximum possible number of function evaluations that are needed for the full cycle. For NLPQL, the number of evaluations can be approximated according to the Finite Difference Approximation gradient calculation method, as follows:

  For **Central**: number of iterations * (2*number of inputs +1)

  For **Forward**: number of iterations * (number of inputs+1)
- **Allowable Convergence (%)**: Stop criterion. Tolerance to which the Karush-Kuhn-Tucker (KKT) optimality criterion is generated during the NLPQL process. A smaller value indicates more convergence iterations and a more accurate (but slower) solution. A larger value indicates fewer convergence iterations and a less accurate (but faster) solution. For a Direct Optimization system, the default percentage value is 0.1. The maximum percentage value is 100. These values are consistent across all problem types because the inputs, outputs, and gradients are scaled during the NLPQL solution.
- **Initial Finite Difference Delta (%)**: Advanced option for specifying the relative variation used to perturb the current point to compute gradients. Used in conjunction with Allowable Convergence (%) to ensure that the delta in NLPQL's calculation of finite differences is large enough to be seen above the noise in the simulation problem. This wider sampling produces results that are more clearly differentiated so that the difference is less affected by solution noise and the gradient direction is clearer. The value should be larger than both the value for Initial Finite Difference Delta (%) and the noise magnitude of the model. However, smaller values produce more accurate results, so set Initial Finite Difference Delta (%) only as high as necessary to be seen above simulation noise.

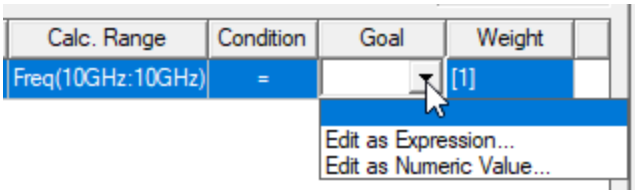
  The default percentage value is 1. The minimum is 0.0001, and the maximum is 1. For parameters with Allowed Values set to **Manufacturable Values** or Snap to Grid, the value for Initial Finite Difference Delta (%) is ignored. In such cases, the closest allowed value is used to determine the finite difference delta.

5. Add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.

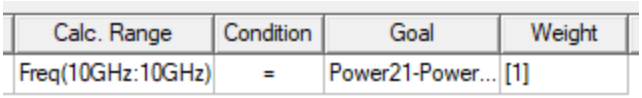


When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the **Add/EditCalculation** dialog.

6. In the Optimization setup, in the dropdown for the Goal column, select either Edit as Expression or Edit as Numeric Value...



7. This reopens the **Add/Edit Calculation** dialog box. If you are satisfied with the expression or value displayed, click **Done** to close the dialog box. This enters the expression/value to the **Goal** column.



6. In the **Optimization** setup, if you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

     The **Cost Function Norm Type** pull-down list appears.
   - Select **L1**, **L2**, or **Maximum**.

     A norm is a function that assigns a positive value to the cost function.

     For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors, which means that it is always less than zero. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.

7. Optionally, set the Acceptable Cost and Cost Function Noise.
8. Optionally, click the button for setting HPC and Analysis Options, which allows you to select or create an analysis configuration.
9. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization.
   - You may also override the variable starting values by clicking the **Override** check box and entering the desired value in the **Starting Value** field.
   - Optionally, modify the values of fixed variables that are not being optimized.
   - Optionally, set Linear constraints.
   - Select the **View all columns** check box to see all columns, including hidden columns.
10. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

    Enabling the **Update design parameters' value after optimization** check box will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.
11. Under the **Options tab**, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note	Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

    You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation. This will provide some speed improvement in the overall optimization process.

Setup Mixed-Integer Sequential Quadratic Programming (Gradient and Discrete) Optimizer

Following is the procedure for setting up an optimization analysis using the Mixed-Integer Sequential Quadratic Programming optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

The Mixed-Integer Sequential Quadratic Programming (Nonlinear Programming by Quadratic Lagrangian) method (MISQP)can be used for Direct Optimization systems. It allows you to generate a new sample set to provide a more refined approach than the Screening method. MISQP is available for both continuous and discrete input parameters, which is why mixed is in its name. MISQP can handle only one output parameter goal.. Other output parameters can be defined as constraints. For more information, see Convergence Rate % and Initial Finite Difference Delta % in NLPQL and MISQP and Nonlinear Programming by Quadratic Lagrangian (NLPQL).

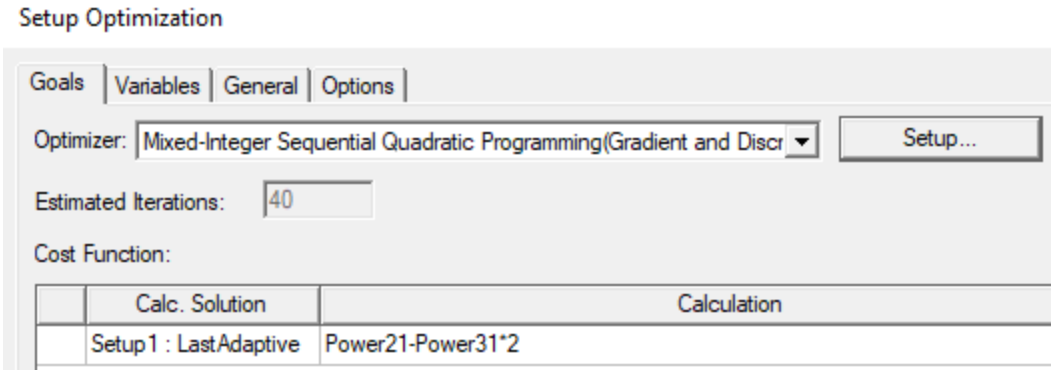
To generate samples and perform an NLPQL optimization:

1. Set up the variables you want to optimize in the **Design Properties** dialog box. The variables must be swept in a Parametric setup.
2. Click **HFSS** or **Q3D Extractor** or **2D Extractor>Optimetrics Analysis>Add**

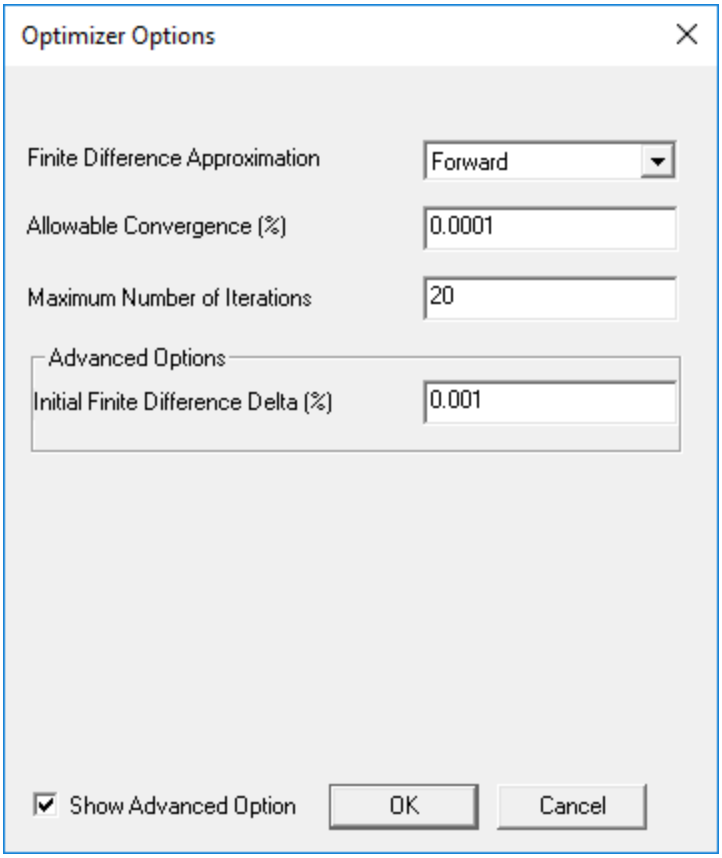
   

   Optimization .

   The **Setup Optimization** dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **Mixed-Integer Sequential Quadratic Programming (Gradient and Discrete)** from the **Optimizer** drop-down menu.



4. Optionally press the **Setup** button to open the **Optimizer Options window**.



- **Finite Difference Approximation**: When analytical gradients are not available, MISQP approximates them numerically. This property allows you to specify the method of approximating the gradient of the objective function. Choices are:
- **Central**: Increases the accuracy of the gradient calculations by sampling from both sides of the sample point but increases the number of design point evaluations by 50%. This method makes use of the initial point, as well as the forward point and rear point.
- **Forward**: Uses fewer design point evaluations but decreases the accuracy of the gradient calculations. This method makes use of only two design points, the initial point and forward point, to calculate the slope forward. This is the default method for new Direct Optimization systems.
- **Maximum Number of Iterations**: Stop criterion. Maximum number of iterations that the algorithm is to execute. If convergence happens before this number is reached, the iterations stop. This also provides an idea of the maximum possible number of function evaluations that are needed for the full cycle. For MISQP, the number of evaluations can be approximated according to the Finite Difference Approximation gradient calculation method, as follows:

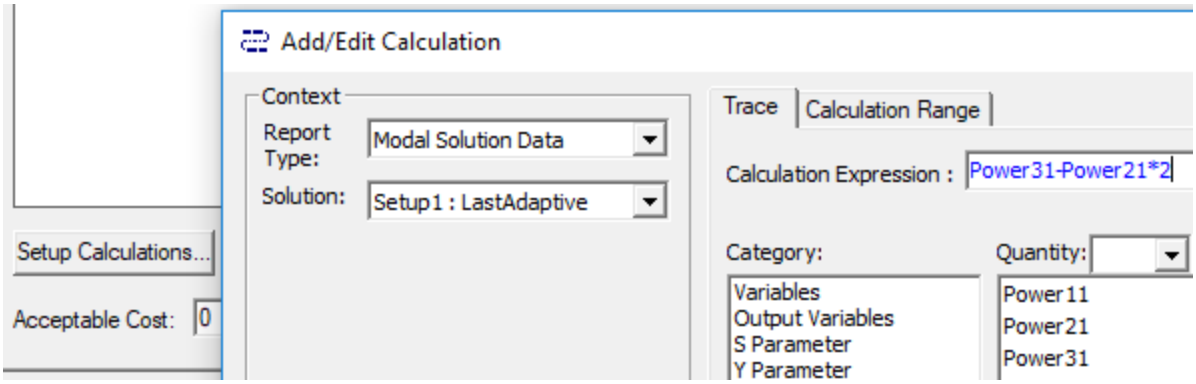
  For **Central**: number of iterations * (2*number of inputs +1)

For **Forward**: number of iterations * (number of inputs+1)

- **Allowable Convergence (%)**: Stop criterion. Tolerance to which the Karush-Kuhn-Tucker (KKT) optimality criterion is generated during the MISQP process. A smaller value indicates more convergence iterations and a more accurate (but slower) solution. A larger value indicates fewer convergence iterations and a less accurate (but faster) solution. For a Direct Optimization system, the default percentage value is 0.1. The maximum percentage value is 100. These values are consistent across all problem types because the inputs, outputs, and gradients are scaled during the MISQP solution.
- **Initial Finite Difference Delta (%)**: Advanced option for specifying the relative variation used to perturb the current point to compute gradients. Used in conjunction with Allowable Convergence (%) to ensure that the delta in MISQP's calculation of finite differences is large enough to be seen above the noise in the simulation problem. This wider sampling produces results that are more clearly differentiated so that the difference is less affected by solution noise and the gradient direction is clearer. The value should be larger than both the value for Initial Finite Difference Delta (%) and the noise magnitude of the model. However, smaller values produce more accurate results, so set Initial Finite Difference Delta (%) only as high as necessary to be seen above simulation noise.

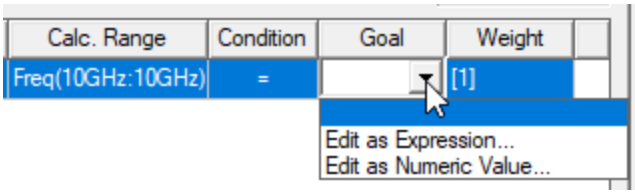
  The default percentage value is 1. The minimum is 0.0001, and the maximum is 1. For parameters with Allowed Values set to **Manufacturable Values** or Snap to Grid, the value for Initial Finite Difference Delta (%) is ignored. In such cases, the closest allowed value is used to determine the finite difference delta.

5. Add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.

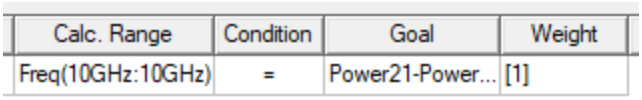


When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the **Add/EditCalculation** dialog.

6. In the Optimization setup, in the dropdown for the Goal column, select either Edit as Expression or Edit as Numeric Value...



This reopens the **Add/Edit Calculation** dialog box. If you are satisfied with the expression or value displayed, click **Done** to close the dialog box. This enters the expression/value to the **Goal** column.



7. In the **Optimization** setup, if you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

     The **Cost Function Norm Type** pull-down list appears.
   - Select **L1**, **L2**, or **Maximum**.

     A norm is a function that assigns a positive value to the cost function.

     For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors, which means that it is always less than zero. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

     The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.
8. Optionally, set the Acceptable Cost and Cost Function Noise.
9. Optionally, click the button for setting HPC and Analysis Options, which allows you to select or create an analysis configuration.
10. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization.
    - You may also override the variable starting values by clicking the **Override** check box and entering the desired value in the **Starting Value** field.
    - Optionally, modify the values of fixed variables that are not being optimized.
    - Optionally, set Linear constraints.
    - Select the **View all columns** check box to see all columns, including hidden columns.
11. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

    Enabling the **Update design parameters' value after optimization** check box will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.

12. Under the **Options tab**, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note	Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation. This will provide some speed improvement in the overall optimization process.

Setup Adaptive Multiple-Objective (Random Search) Optimizer

Following is the procedure for setting up an optimization analysis using the Adaptive Multiple Objective (Random Search) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

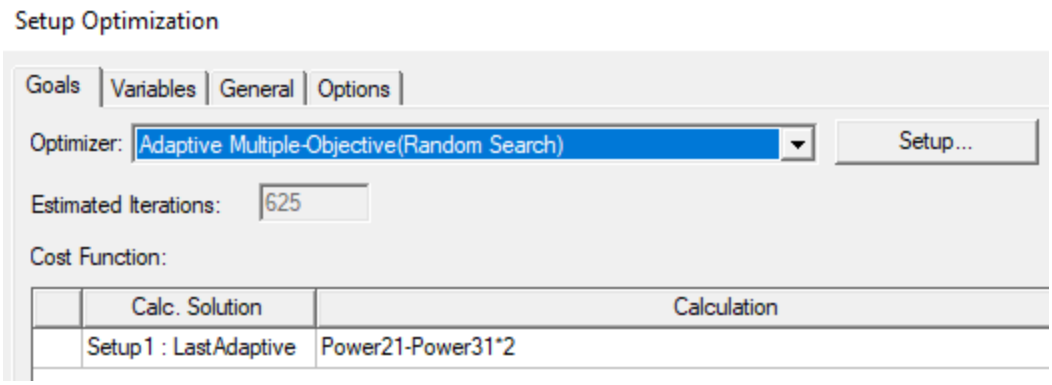
The Adaptive Multiple Objective (Kriging + MOGA) is an iterative algorithm that allows you to either generate a new sample set or use an existing set, providing a more refined approach than the Screening method. It uses the same general approach as MOGA, but applies the Kriging error predictor to reduce the number of evaluations needed to find the global optimum. The Adaptive Multiple-Objective method is available only for continuous input parameters, including those with manufacturable values. It can handle multiple objectives and multiple constraints. For more information, see Adaptive Multiple-Objective Optimization.

1. Set up the variables you want to optimize in the **Design Properties** dialog box. The variables must be swept in a Parametric setup.
2. Click **HFSS** or **Q3D Extractor** or **2D Extractor>Optimetrics Analysis>Add**

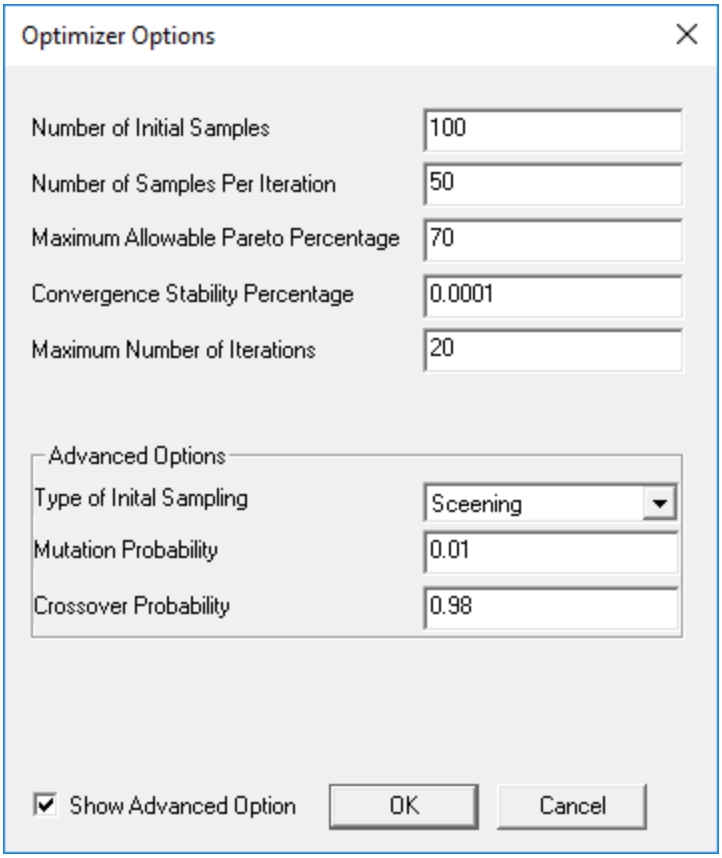
   

   Optimization .

   The **Setup Optimization** dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **Adaptive Multiple Objective (Random-search)** from the **Optimizer** drop-down menu.



4. Optionally press the **Setup** button to open the **Optimizer Options window**.



- **Number of Initial Samples**: Initial number of samples to use. This number must be greater than the number of enabled input parameters. The minimum recommended number of initial samples is 10 times the number of enabled input parameters. The larger the initial sample set, the better your chances of finding the input parameter space that contains the best solutions.

  The number of enabled input parameters is also the minimum number of samples required to generate the Sensitivities chart. You can enter a minimum of 2 and a maximum of 10000. The default is 100.

  If you switch from the Screening method to the MOGA method, MOGA generates a new sample set. For the sake of consistency, enter the same number of initial samples as you used for the Screening method.
- **Number of Samples Per Iteration**: Number of samples to iterate and update with each iteration. This number must be greater than the number of enabled input parameters but less than or equal to the number of initial samples. The default is for a Direct Optimization system.

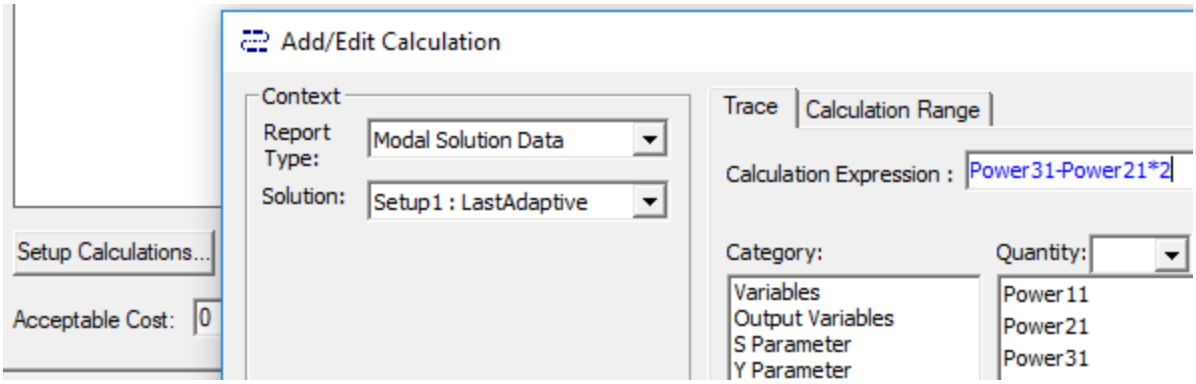
  You can enter a minimum of 2 and a maximum of 10000.
- **Maximum Allowable Pareto Percentage**: Convergence criterion. Percentage value that represents the ratio of the number of desired Pareto points to the number of samples per iteration. When this percentage is reached, the optimization is converged. For

example, a value of 70 with Number of Samples Per Iteration set to 200 would mean that the optimization should stop once the resulting front of the MOGA optimization contains at least 140 points. Of course, the optimization stops before that if the maximum number of iterations is reached.

- If the **Maximum Allowable Pareto Percentage** is too low (below 30), the process can converge prematurely. If the value is too high (above 80), the process can converge slowly. The value of this property depends on the number of parameters and the nature of the design space itself. The default is 70. Using a value between 55 and 75 works best for most problems. For more information, see Convergence Criteria in MOGA-Based Multi-Objective Optimization .
- **Convergence Stability Percentage**: Convergence criterion. Percentage value that represents the stability of the population based on its mean and standard deviation. This criterion allows you to minimize the number of iterations performed while still reaching the desired level of stability. When the specified percentage is reached, the optimization is converged. The default percentage is 2. To not take the convergence stability into account, set to 0. For more information, see Convergence Criteria in MOGA-Based Multi-Objective Optimization .
- **Maximum Number of Iterations**: Stop criterion. Maximum number of iterations that the algorithm is to execute. If this number is reached without the optimization having reached convergence, iterations stop. This also provides an idea of the maximum possible number of function evaluations that are needed for the full cycle, as well as the maximum possible time it can take to run the optimization. For example, the absolute maximum number of evaluations is given by:

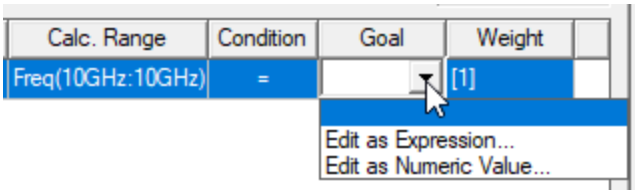
  Number of Initial Samples + Number of Samples Per Iteration * (Maximum Number of Iterations - 1)
- **Type of Initial Sampling**: Advanced option for generating different kinds of sampling. If you do not have any parameter relationships defined, set to Screening (default) or Optimal Space-Filling. If you have parameter relationships defined, the initial sampling must be performed by the constrained sampling algorithms because parameter relationships constrain the sampling. For such cases, this property is automatically set to Constrained Sampling.
- **Mutation Probability**: Advanced option for specifying the probability of applying a mutation on a design configuration. The value must be between 0 and 1. A larger value indicates a more random algorithm. If the value is 1, the algorithm becomes a pure random search. A low probability of mutation (<0.2) is recommended. The default is 0.01. For more information on mutation, see MOGA Steps to Generate a New Population.
- **Crossover Probability**: Advanced option for specifying the probability with which parent solutions are recombined to generate offspring solutions. The value must be between 0 and 1. A smaller value indicates a more stable population and a faster (but less accurate) solution. If the value is 0, the parents are copied directly to the new population. A high probability of crossover (>0.9) is recommended. The default is 0.98.

5. Add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.

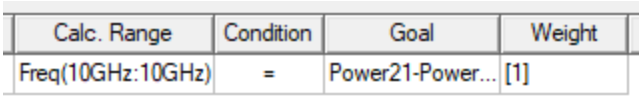


When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the **Add/EditCalculation** dialog.

6. In the Optimization setup, in the dropdown for the Goal column, select either **Edit as Expression...** or **Edit as Numeric Value...**.



This reopens the **Add/Edit Calculation** dialog box. If you are satisfied with the expression or value displayed, click **Done** to close the dialog box. This enters the expression/value to the **Goal** column.



7. In the **Optimization** setup, if you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

     The **Cost Function Norm Type** pull-down list appears.
   - Select **L1**, **L2**, or **Maximum**.

     A norm is a function that assigns a positive value to the cost function.

     For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors, which means that it is always less than zero. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

     The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.

8. Optionally, set the Acceptable Cost and Cost Function Noise.
9. Optionally, click the button for setting HPC and Analysis Options, which allows you to select or create an analysis configuration.
10. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization.
    - You may also override the variable starting values by clicking the **Override** check box and entering the desired value in the **Starting Value** field.
    - Optionally, modify the values of fixed variables that are not being optimized.
    - Optionally, set Linear constraints.
    - Select the **View all columns** check box to see all columns, including hidden columns.
11. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

    Enabling the **Update design parameters' value after optimization** check box will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.
12. Under the **Options tab**, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note	Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

    You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation. This will provide some speed improvement in the overall optimization process.

Setup Adaptive Single-Objective (Gradient) Optimizer

Following is the procedure for setting up an optimization analysis using the Adaptive Single-Objective (OSF + Kriging + MISQP) Optimizer. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes.

This gradient-based method employs automatic intelligent refinement to provide the global optima. It requires a minimum number of design points to build the Kriging response surface, but, in general, this method reduces the number of design points necessary for the optimization. Failed design points are treated as inequality constraints, making it fault-tolerant.

The Adaptive Single-Objective method is available for input parameters that are continuous, including those with manufacturable values. It can handle only one output parameter goal, although other output parameters can be defined as constraints. It does not support the use of parameter relationships in the optimization domain. For more information, see Adaptive Single-Objective Optimization (ASO). It requires advanced options. Ensure that the Show Advanced Options check box is selected.

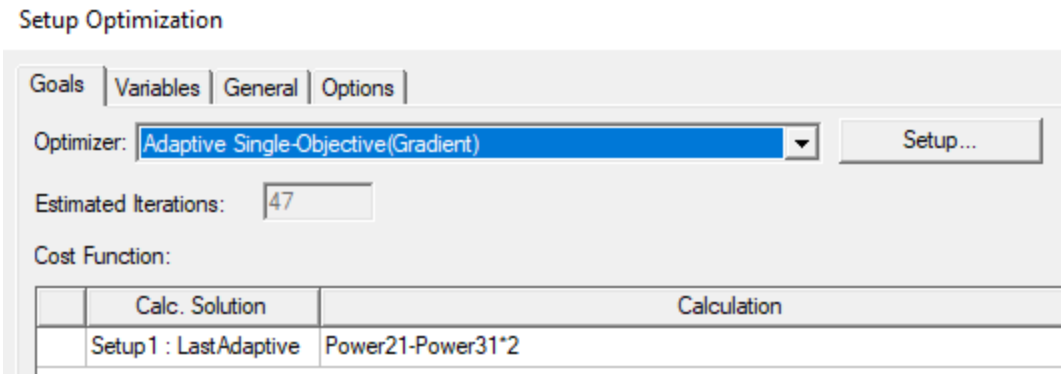
1. Set up the variables you want to optimize in the **Design Properties** dialog box. The variables must be swept in a Parametric setup.

2. Click **HFSS** or **Q3D Extractor** or **2D Extractor>Optimetrics Analysis>Add**

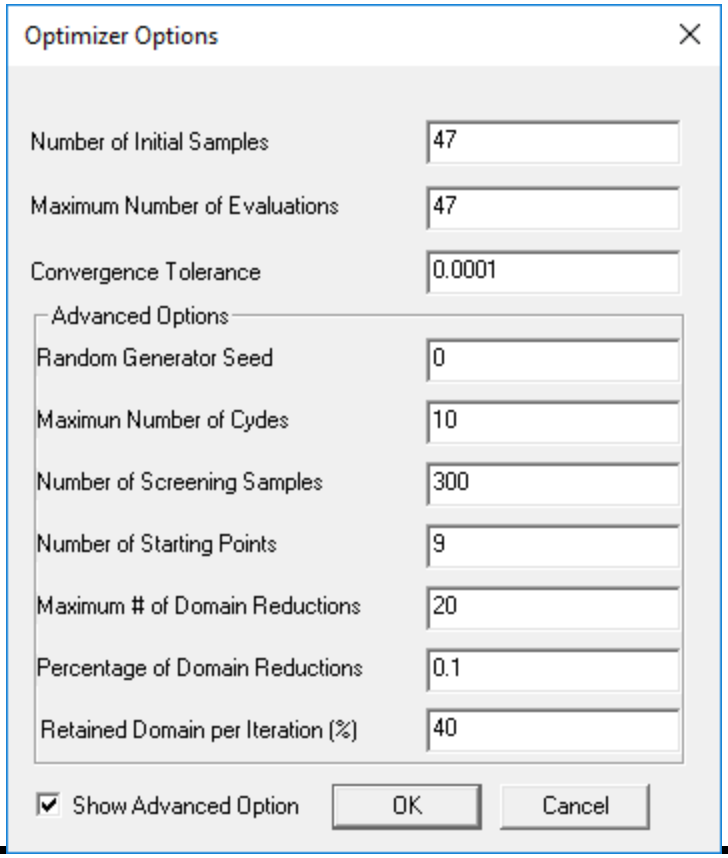
   

   Optimization .

   The **Setup Optimization** dialog box appears.
3. Under the **Goals** tab, select the optimizer by selecting **Adaptive Single Objective (Gradient)** from the **Optimizer** drop-down menu.



4. Optionally press the **Setup** button to open the **Optimizer Options** window and check Advanced Options.



  - **Number of Initial Samples**: Number of samples generated for the initial Kriging and after all domain reductions for the construction of the next Kriging. You can enter a minimum of (NbInp+1)*(NbInp+2)/2 (also the minimum number of OSF samples required for the Kriging construction) or a maximum of 10,000. The default is (NbInp+1)* (NbInp+2)/2.

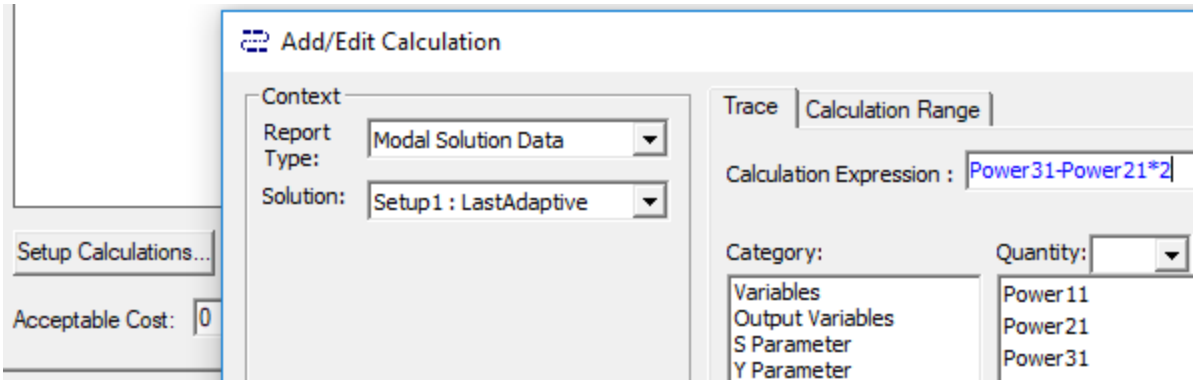
    Because of the Adaptive Single-Objective workflow (in which a new OSF sample set is generated after each domain reduction), increasing the number of OSF samples does not necessarily improve the quality of the results and significantly increases the number of evaluations
  - **Maximum Number of Evaluations**:Stop criterion. Maximum number of evaluations (design points) that the algorithm is to calculate. If convergence occurs before this number is reached, evaluations stop. This value also provides an idea of the maximum possible time it takes to run the optimization. The default is 20*(NbInp +1).
- **Convergence Tolerance**: Stop criterion. Minimum allowable gap between the values of two successive candidates. If the difference between two successive candidates is smaller than the value for Convergence Tolerance multiplied by the maximum variation of the parameter, the algorithm is stopped. A smaller value indicates more convergence iterations and a more accurate (but slower) solution. A larger value indicates fewer convergence iterations and a less accurate (but faster) solution. The default is 1E-06.
- **Random Generator Seed**: The value for initializing the random number generator invoked internally by OSF. The value must be a positive integer. This property allows you to generate different samplings by changing the value or to regenerate the same sampling by keeping the same value. The default is 0.
- **Maximum Number of Cycles**: Number of optimization loops that the algorithm needs, which in turns determines the discrepancy of the OSF. The optimization is essentially combinatorial, so a large number of cycles slows down the process. However, this makes the discrepancy of the OSF smaller. The value must be greater than 0. For practical purposes, 10 cycles is usually good for up to 20 variables. The default is 10.
- **Number of Screening Samples**: Number of samples for the screening generation on the current Kriging. This value is used to create the next Kriging (based on error prediction) and verified candidates.

    You can enter a minimum of (NbInp+1)*(NbInp+2)/2 (also the minimum number of OSF samples required for the Kriging construction) or a maximum of 10,000. The default is 100*NbInp for a Direct Optimization system. There is no default for a Response Surface Optimization system.

    The larger the screening sample set, the better the chances of finding good verified points. However, too many points can result in a divergence of the Kriging.
- **Number of Starting Points**: Determines the number of local optima to explore. The larger the number of starting points, the more local optima explored. In the case of a linear surface, for example, it is not necessary to use many points. This value must be less than the value for **Number of Screening Samples** because these samples are selected in this sample. The default is the value for **Number of Initial Samples**.

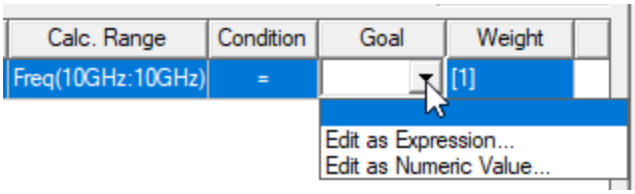
- **Maximum Number of Domain Reductions**: Stop criterion. Maximum number of domain reductions for input variation. (No information is known about the size of the reduction beforehand.) The default is 20.
- **Percentage of Domain Reductions**: Stop criterion. Minimum size of the current domain according to the initial domain. For example, with one input ranging between 0 and 100, the domain size is equal to 100. The percentage of domain reduction is 1%, so the current working domain size cannot be less than 1 (such as an input ranging between 5 and 6). The default is 0.1.
- **Retained Domain per Iteration (%)**: Advanced option that allows you to specify the minimum percentage of the domain you want to keep after a domain reduction. The percentage value must be between 10 and 90. A larger value indicates less domain reduction, which implies better exploration but a slower solution. A smaller value indicates a faster and more accurate solution, with the risk of it being a local one. The default percentage value is 40.

5. Add a cost function by selecting the **Setup Calculations** button to open the **Add/Edit Calculation** dialog.



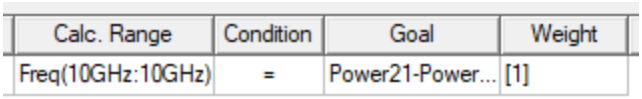
When you have created the calculation, click **Add Calculation** to add it to the **Optimization** setup, and **Done** to close the **Add/EditCalculation** dialog.

6. In the Optimization setup, in the dropdown for the Goal column, select either **Edit as Expression...** or **Edit as Numeric Value...**.



This reopens the **Add/Edit Calculation** dialog box. If you are satisfied with the expression or value displayed, click **Done** to close the dialog box. This enters the expression/value to

the **Goal** column.



Calc. Range	Condition	Goal	Weight
Freq(10GHz:10GHz)	=	Power21-Power...	[1]

7. In the **Optimization** setup, if you want to select a **Cost Function Norm Type**:
   - Check the **Show Advanced Option** check box.

     The **Cost Function Norm Type** pull-down list appears.
   - Select **L1**, **L2**, or **Maximum**.

     A norm is a function that assigns a positive value to the cost function.

     For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors. For **L2** norm (the default) the actual cost function uses the weighted sum of squared values of the individual goal error. For the Maximum norm the cost function uses the maximum among all the weighted goal errors, which means that it is always less than zero. (For further details, see Explanation of the L1, L2, and Max Norms in Optimization.)

     The norm type doesn't impact goal setting that use as condition the "minimize" or "maximize" scenarios.
8. Optionally, set the Acceptable Cost and Cost Function Noise.
9. Optionally, click the button for setting HPC and Analysis Options, which allows you to select or create an analysis configuration.
10. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization.
    - You may also override the variable starting values by clicking the **Override** check box and entering the desired value in the **Starting Value** field.
    - Optionally, modify the values of fixed variables that are not being optimized.
    - Optionally, set Linear constraints.
    - Select the **View all columns** check box to see all columns, including hidden columns.
11. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.

    Enabling the **Update design parameters' value after optimization** check box will cause Optimetrics to modify the variable values in the nominal design to match the final values from the optimization analysis.
12. Under the **Options tab**, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note	Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

    You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation. This will provide some speed improvement in the overall optimization process.

Setting the Maximum Iterations for an Optimization Analysis

The **Max. No. of Iterations** value is the maximum number of design variations that you want Optimetrics to solve during an optimization when using the **SNLP**, **SMINLP**, **Quasi Newton**, or **Pattern Search Optimizer**. This value is a stopping criterion; if the maximum number of iterations has been completed, the optimization analysis stops. If the maximum number of iterations has not been completed, the optimization continues by performing another iteration, that is, by solving another design variation.

If the maximum number of iterations has not been reached, the optimizer performs iterations until the acceptable cost function is reached or until the optimizer cannot proceed as a result of other optimization setup constraints, such as when it searches for a variable value with a step size smaller than the minimum step size.

Note	The **Genetic Algorithm** optimizer does not use the **Max. No. of Iterations** criteria.

To set the maximum number of iterations for an optimization analysis:

- Under the **Goals** tab of the **Setup Optimization** dialog box, type a value in the **Max. No. of Iterations** text box.

Related Topics

Adding a Cost Function

Cost Function

Optimetrics manipulates the model's design variable values to find the minimum location of the cost function; therefore, you should define the cost function so that a minimum location is also the optimum location.

When using the Quasi Newton optimizer, which is appropriate for designs that are not sensitive to noise, the best cost function is a smooth, second-order function that can be approximated well by quadratics in the vicinity of the minimum; the slope of the cost function should decrease as Optimetrics approaches the optimum value. The preferred cost function takes values between 0 and 1. In practice, most functions that are smooth around the minimum are acceptable as cost functions. Most importantly, the cost function should not have a sharp dip or pole at the minimum. A well designed cost function can significantly reduce the optimization process time.

The cost function is defined in the **Setup Optimization** dialog box or the **Design of Experiments** setup when you set up an optimization analysis. If you know the exact syntax of the solution quantity on which you want to base the cost function, you can type it directly in the **Calculation** text box. You can also use **Setup Calculations** to add a solution quantity via the **Add/Edit Calculation** dialog box, or to create an output variable that represents the solution quantity in the Output Variables dialog box.

Related Topics

Adding a Cost Function
Acceptable Cost
Cost Function Noise
Linear Constraints
Goal Weight
Step Size
Explanation of L1, L1, Norm Costs in Optimization

Acceptable Cost

The acceptable cost is the value of the cost function at which the optimization process should stop; otherwise known as the *stopping criterion*.The cost function value must be equal to or below the acceptable cost value for the optimization analysis to stop. The acceptable cost may be a negative value.

Related Topics

Cost Function
Adding a Cost Function

Cost Function Noise

The numerical calculation of the electromagnetic field introduces various sources of noise to the cost function, particularly because of changes in the finite element mesh. You must provide the optimizer with an estimate of the noise. The noise indicates whether a change during the solution process is significant enough to support achievement of the cost function.

For example, if the cost function, *c*, is

$$c = 10000 \cdot \left| L_{11} \right|^2$$

where $|L_{11}|$ is the magnitude of the inductance, at the minimum, $|L_{11}|$ is expected to be very small,

$$\left| L_{11} \right| \approx 0$$

.

From the solution setup, the error in $|L_{11}|$ is expected to be $E_{L11} \approx 0.01$. The perturbed cost function is therefore

$$c_{perturbed} = 10000 \cdot (|L_{11}|_{min} + E_{L11})^2$$

Near the minimum, the error in the cost function E_c is given by

$$E_c = c_{perturbed} - c_{min} = 10000 \cdot (0.0 + 0.01)^2 - (10000 \cdot 0.0) = 1.0$$

Therefore, the cost function noise would be 1.0.

Related Topics

Cost Function

Adding a Cost Function

A cost function can include one or more goals for an optimization analysis. Optimetrics manipulates the model's design variable values to fulfill the cost function.The optimization will stop when the solution quantity meets the acceptable cost criterion.

Following is the general procedure for adding a cost function with a single goal:

1. Under the **Goals** tab of the **Setup Optimization** dialog box, click **Setup Calculations...**
   The Add/Edit Calculation dialog box is displayed.
2. In the **Add/Edit Calculation** dialog box, follow these general steps to set up a cost function.
   a. Set the **Context** for the calculation.
   b. Choose the **Category** of available data type depending upon the Solution type of the design being optimized.
   c. Select the **Quantity** to add to the **Calculated Expression** field. Available quantities depend upon the **Category** selection.
   d. You may optionally make a selection from the function list to apply to the calculated expression.
   e. When the **Calculation Expression** has the desired equation, click **Add Calculation** to add the expression to the cost function table.

f. Repeat to add additional calculations to the cost function or click **Done** to exit the **Add/Edit Calculation** dialog box and return to **Setup Optimization**.

3. To modify the **Solution** on which the calculation is based, click in the **Solution** column and select the solution from which the cost function is to be extracted from the pull-down list.
4. To edit the calculation on which to base the cost function goal, select **Edit** from the pull-down list.
5. In the **Condition** text box, click one of the following conditions from the pull-down list:

<=	Less than or equal to
=	Equal to
>=	Greater than or equal to
Minimize	Reduce the cost function to a minimum value
Maximize	Identify a maximized condition

6. In the **Goal** text box, type the value of the solution quantity that you want to be achieved during the optimization analysis. If the solution quantity is a complex calculation, the goal value must be complex; two goal values must be specified. The **Minimize** and **Maximize** options do not require you to specify a **Goal** value.
7. Optionally, if you have multiple goals and want to assign higher or lower priority to a goal, type a different value for the goal's weight in the **Weight** text box. The goal with the greater weight is given more importance. If the goal is a complex value, the weight value must be complex; two weight values must be specified. The weight value cannot be variable dependent.

Note	Click the **Edit Goal/Weight** button to open the **Edit Goal Value/Weight** dialog box where you can modify weights for all goals simultaneously; as well as, set the **Goal Values** to expressions.

8. Optionally, click the button for setting **HPC and Analysis Options**, which allows you to select or create an analysis configuration.
9. Specify other options (such as acceptable cost, noise, and number of passes), and then click **OK**.

The optimization stops when the solution quantity meets the acceptable cost criterion.

Related Topics

Setting a Goal Value
Cost Function
Acceptable Cost
Goal Weight

Adding/Editing a Cost Function Calculation

The Add/Edit Calculation dialog box allows you to define the mathematical equation for one or multiple cost functions. It represents the calculation to be performed on the optimization variables to compare to the goal values. To set up a calculation for a cost function:

1. In the **Context** section of the dialog:
   - Select the **Report Type** with a pull-down selection list containing the available types for this design.
   - Select the **Solution** from the drop down selection list. This lists the available setups and sweeps. As a minimum, the **LastAdaptive** solution is available.
   - Select the Geometry from the drop down selection list or select none (the default). This modifies the list of quantities available to the ones that apply to the specific geometry.
2. The **Output Variables** button opens the Output Variables dialog box allowing you to create special output variables to be used in the cost function.
3. The **Calculated Expression** field in the **Trace** tab is used to enter the equation to be used for the cost function. To enter an expression, you may type it directly into the field or use the **Category**, **Quantity**, and **Function** lists as follows:
   - Select the **Category**, these depend on the Solution type and the design. This lets you specify the category of information to be used in the cost function.
   - Select a **Quantity** from the list. Available quantities depend upon the Solution type, as well as the Geometry and Category selection. Selecting a Quantity automatically enters it into the Calculated Expression field.
   - Select a **Function** to apply to the value in the calculated expression.
   - For swept variables, the Range **Function** button opens the **Set Range Function** dialog to apply functions to the expression that apply over the sweep range.
4. The **Calculation Range** tab applies to swept variables and allows you to specify the range of the sweep over which to apply the calculation.
5. When the desired **Calculated Expression** has been obtained, click the **Add Calculation** button to add the entry to the cost function table. You may add multiple entries to the table simply by changing the **Calculated Expression** and using the **Add Calculation** button.
6. To update or edit a selected cost function, enter the desired Calculated Expression and click the **Update Calculation** button.
7. Click **Done** to return to the Setup Calculations dialog box.

Specifying a Solution Quantity for a Cost Function Goal

When setting up a cost function, you must identify the solution quantity on which to base each goal. Solution quantities are specified by mathematical expressions that are composed of basic quantities, such as matrix parameters, and output variables.

1. Add a row (a goal) to the cost function table:
   a. Under the **Goals** tab of the **Setup Optimization** dialog box, click **Add**.
      A new row is added to the **Cost Function** table.

b. In the **Solution** column, click the solution from which the cost function is to be extracted.

2. In the **Solution** text box, click the solution from which the solution quantity is to be extracted.
3. In the **Calculation** text box, specify the solution quantity in one of the following ways:
   - If you know the syntax of the mathematical expression or the output variable's name, type it in the **Calculation** text box.
   - If you want to create an output variable that represents the solution quantity, do the following:
     a. Click **Edit Calculation**.
        The **Output Variables** dialog box appears.
     b. Add the expression you want to evaluate, click **Done**.
     c. Click **Done** to close the **Output Variables** dialog box.
        In the **Setup Optimization** dialog box, the most recently created output variable appears in the **Calculation** text box.
     d. To specify a different defined output variable, click the **Calculation** text box. It becomes a pull-down list that displays all of the defined output variables. Click an output variable from the pull-down list.

Setting the Calculation Range of a Cost Function Goal

The calculation range is the range within which you want a cost function goal to be calculated. It can be a single value or a range of values, depending on the solution or solution quantity selected for the goal.

1. Under the **Goals** tab in the **Setup Optimization** dialog box, click **Edit Cal. Range**.
2. In the **Variable** pull-down list, click a variable.

   If you chose to solve a parametric setup during the optimization analysis, the variables swept in that parametric setup are available in the **Variable** pull-down list. If you sweep a variable in the parametric setup that is also being optimized, that variable is excluded from the optimization.

   Other examples of available variables include frequency, if the solution quantity is an S-parameter quantity, and phi or theta, if the solution quantity is a radiated field quantity.
3. After you select a variable from the **Variable** pull-down list, you can select a range of values for the calculation range as follows:
   a. Select **Range**.
   b. In the **Start** text box, type the starting value of the range.
   c. In the **Stop** text box, type the final value of the range.
4. To select a single value for the calculation range:
   a. Select **Single Value**.
   b. In the **Value** text box, type the value of the variable at which the cost function goal is to be extracted.
5. Click **Update**, and then click **OK**.

Setting a Goal Value

A goal is the value you want a solution quantity to reach during an optimization analysis. It can be a real value or a complex value. If the solution quantity is a complex calculation, the goal value must be complex. You can type the goal value in the **Goal** text box. Alternatively, you can use the **Edit Goal/Value Weight** dialog box to specify the goal value as a single value, a mathematical expression, or a value dependent on a variable such as frequency.

Related Topics

Specify a single goal value.
Specify an expression as the goal value.
Specify a variable-dependent goal value.

Specifying a Single Goal Value

1. Under the **Goals** tab in the **Setup Optimization** dialog box, click **Edit Goal/Weight**.

   The **Edit Goal/Weight** dialog box appears.

2. Under the **Goal Value** tab, click **Simple Numeric Value** from the **Type** list.
3. If the goal value is complex, click **real/imag** in the pull-down list to the right if you want to specify the real and imaginary parts of the goal value.

   Alternatively, click **mag/ang** if you want to specify the magnitude and angle of the goal value.

4. Type the goal value in the **Goal Value** table.

   If the goal value is complex, type both parts of the goal value in the text box below the **Goal Value** heading. For example, type **1, 1** to specify the real part of the goal value as 1 and the imaginary part as 1.

   If the goal value is real, type a real goal value in the text box below the **Goal Value** heading.

5. Click **OK**.

   The goal value you specified appears in the **Goal** text box.

Specifying an Expression as a Goal Value

1. Under the **Goals** tab in the **Setup Optimization** dialog box, click **Edit Goal/Weight**.

   The **Edit Goal/Weight** dialog box appears.

2. Under the **Goal Value** tab, click **Expression** from the **Type** list.
3. If you know the syntax of the mathematical expression or the existing output variable's name, type it in the text box below the **Goal Value** heading.

   Alternatively, if you want to create an output variable that represents the goal value, do the following:

   a. Click **Edit Expression**.
   The **Output Variables** dialog box appears.
   b. Add the expression you want to be the goal value, and then click **Done**.
   Maxwell enters the most recently created output variable in the text box below the **Goal Value** heading.
4. Click **OK**.

   The goal value you specified appears in the **Goal** text box.

Specifying a Variable-Dependent Goal Value

1. Under the **Goals** tab in the **Setup Optimization** dialog box, click **Edit Goal/Weight**.

   The **Edit Goal/Weight** dialog box appears.
2. Under the **Goal Value** tab, click **Variable Dependent** from the **Type** list.
3. Click a variable from the pull-down list to the left of the table.
4. Type the value of that variable in the first column of the table.

Warning	Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

5. Type a corresponding goal value for that variable value in the text box below the **Goal Value** heading.
6. Click **Add** to add another row to the reference curve.
7. Repeat steps 4, 5, and 6 until you have specified the reference curve.
8. Click **OK**.

   The goal value is listed as being variable dependent in the **Goal** text box.

Goal Weight

If an optimization setup has a cost function made up of multiple goals, you can assign a different weight to each goal. The goal with the greater weight is given more importance during the cost calculation.

The error function value is a weighted sum of the sub-goal errors. Each sub-goal, at each frequency at which it is evaluated, gives rise to a (positive) error value that represents the

discrepancy between the simulated response and the goal value limit. If the response satisfies the goal value limit, then the error value is 0. Otherwise, the error value depends on the differences between the simulated response and the respective goal limit. The error function may be defined as follows:

$$\sum_{j}^{G} \frac{W_j}{N_j} \cdot \sum_{i}^{N_j} e_i$$

where

- G is the number of sub-goals.
- W_j is the weight factor associated with the j^{th} sub-goal.
- N_j is the number of frequencies for the j^{th} sub-goal.
- e_i is the error contribution from the j^{th} sub-goal at the i^{th} frequency.

The value of e_i is determined by the band characteristics, target value, and the simulated response value. The choices for band characteristics are <=, =, and >=.

Band Characteristics (Condition)	**e_i evaluation where s_i is the simulated response and g_i is the desired limit.**
<=	$e_i = \begin{cases} 0 & s_i \le g_i \\ s_i - g_i & s_i > g_i \end{cases}$
=	$e_i = \lvert s_i - g_i \rvert$

>=	$e_i = \begin{cases} 0 & s_i \geq g_i \\ g_i - s_i & s_i < g_i \end{cases}$

If the total error value is within the acceptable cost, the optimization stops.

Related Topics

Adding a Cost Function

Cost Function

Modifying the Starting Variable Value for Optimization

A variable's starting value is the first value to be solved during the optimization analysis. Optimetrics automatically sets the starting value of a variable to be the current value set for the nominal design. You can modify this value for each optimization setup.

Note	If you choose to solve a parametric setup before an optimization analysis, a variable's starting value is ignored if a more appropriate starting value is calculated for it during the parametric analysis.

1. In the **Setup Optimization** dialog box, click the **Variables** tab.

   All of the variables that were selected for the optimization analysis are listed.
2. Type a new value in the **Starting Value** text box for the value you want to override, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered is used for this optimization analysis, and the current value set for the nominal model is ignored.
   - Alternatively, you can select the **Override** option first, and then type a new variable value in the **Starting Value** text box.
3. Optionally, click a new unit system in one of the **Units** text boxes.

Note	To revert to the default starting value, clear the **Override** check box.

Related Topics

Setting the Min. and Max. Variable Values for Optimization

Step Size

Setting the Min and Max Focus

Modifying the Starting Variable Value for Sensitivity Analysis

Modifying the Starting Variable Value for Statistical Analysis

Setting the Min. and Max. Variable Values for Optimization

For every optimization setup, Optimetrics automatically sets the minimum and maximum values it will consider for a variable being optimized. Optimetrics sets a variable's minimum value equal to approximately 50% of its starting value. (The starting value is the variable's current value set for the nominal design.) Optimetrics sets the variable's maximum value equal to approximately 150% of the starting value. During the optimization analysis, variable values that lie outside of this range are not considered.

Warning	Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

Related Topics

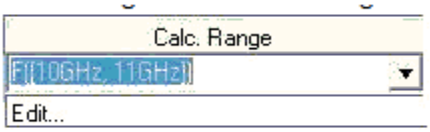
Text Entry for Calc. Range or Edit Calculation Range Dialog

Override the default min and max variable values for a single optimization setup.

Change the default min and max variable values for every optimization setup.

Text Entry for Calc. Range or Edit Calculation Range Dialog

In the **Setup Optimization** dialog, you can enter the Calc. Range Sweep Min/Max by directly editing the Calc. Range field or by accessing an **Edit Calculation Range** dialog.



The edit field accepts the following forms of text:

- sweep that allows you to select different discrete values:

  discrete values, for example, F(10GHz, 11GHz)

  min/max range, for example, F([10GHz, 11GHz])
- editable sweep, which allows you to customize values (that is, a sweep that has an enabled "edited" radio button in sweep selection dialog):

The min/max is used on top of selected values. For example, if you use the sweep dialog and choose "0 deg, 60 deg, 180 deg, 240 deg", then [60deg, 240deg] will select values "60 deg, 180 deg, 240 deg".

- sweep that uses a full range:

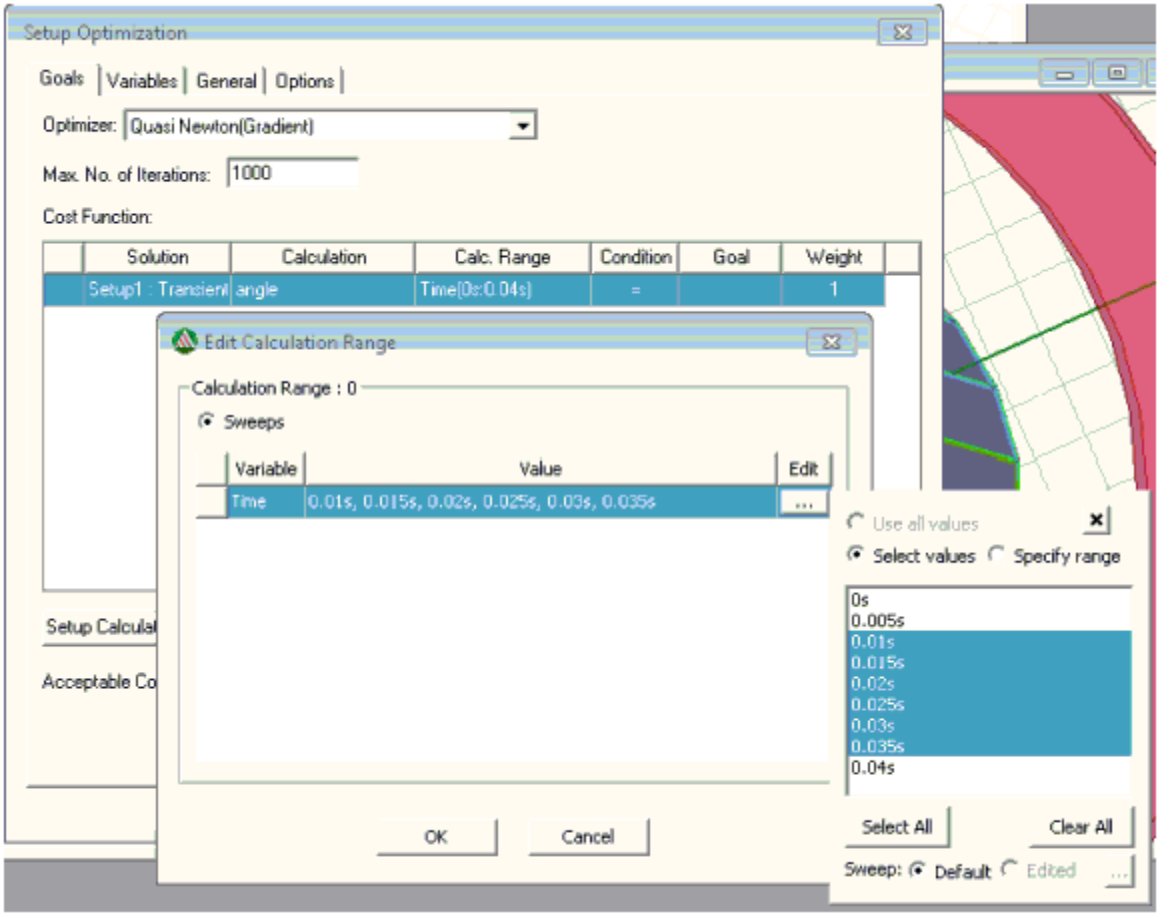
  all values, for example, Time(All)

  min/max range, for example, Time([1ms, 2ms])
- You solve 1 to 20 GHz step .1 and specify F[10.381GHz, 11.381GHz]: it is equivalent to selecting values between 10.4GHz and 11.3GHz.
- You can specify multiple sweep values by separating those with a comma (,).

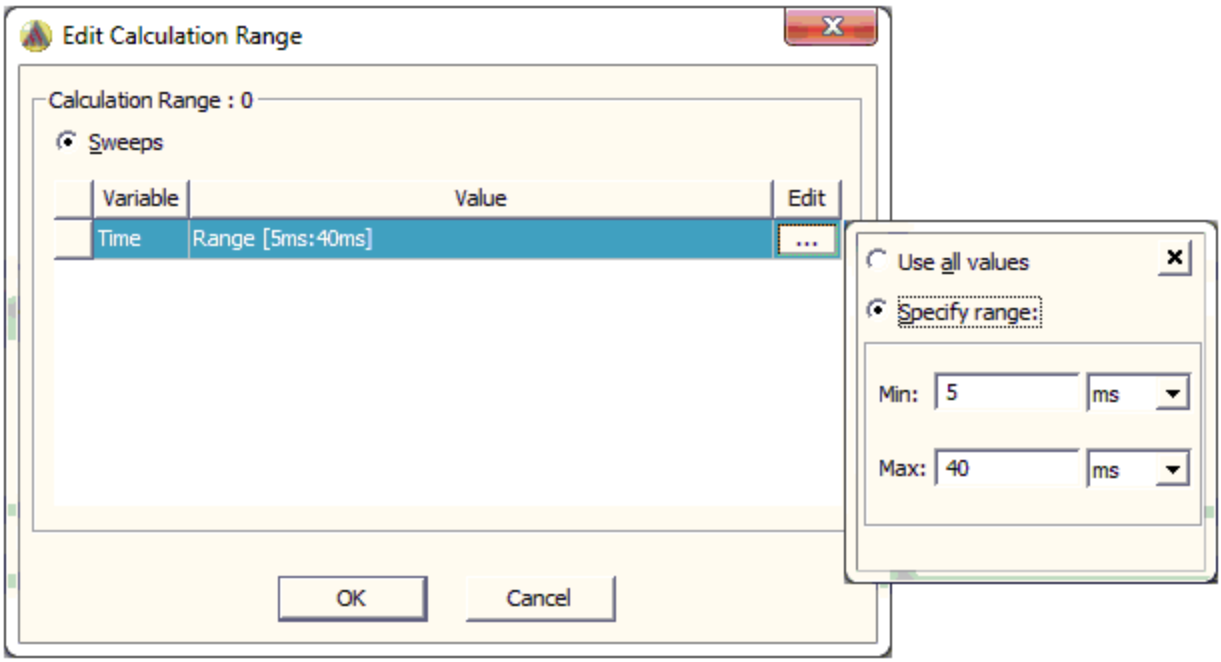
  For example, F(1GHz), cap(1pf, 1.2pf)

  For example, Distance(All), Freq([1ghz,2ghz]), Phase(0 deg)

If you click **Edit** on the menu, you see the **Edit Calculation Range** dialog box. Click on the ellipsis [...] button to select Use all values, Select values, or Specify range. The Select values option is available depending on the solution type.



This example shows how the range appears in the Calc. Range field when you specify a range,.



You could also enter the range directly in the Cal. Range field.

Overriding the Min. and Max. Variable Values for a Single Optimization Setup

1. In the **Setup Optimization** dialog box, click the **Variables** tab.

   All of the variables that were selected for optimization analysis are listed.
2. Type a new value in the **Min** or **Max** text box for the value you want to override, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered is used for this optimization analysis; the variable's current **Min** or **Max** value in the nominal design is ignored.
   - Alternatively, you can select the **Override** option first, and then type a new value in the **Min** or **Max** text box.
3. Optionally, click a new unit system in one of the **Units** text boxes.

To revert to the default minimum and maximum values, clear the **Override** option.

Changing the Min. and Max. Variable Values for Every Optimization Setup

1. Make sure that the variable's minimum and maximum values are not being overridden in any single optimization setup.
2. If the variable is a design variable, do the following: Click **Maxwell3D or Maxwell2D>Design Properties**.

If the variable is a project variable, do the following: Click **Project**>**Project Variables**.

The **Properties** dialog box appears.

3. Select **Optimization**.
4. Type a new value in the **Min** or **Max** text box for the value you want to override, and then press **Enter**.
5. Click **OK**.

   When Optimetrics solves an optimization setup, it does not consider variable values that lie outside of this range.

Step Size

To make the search for the minimum cost value reasonable, the search algorithm is limited in two ways. First, you do not want the optimizer to continue the search if the step size becomes irrelevant or small. This limitation impacts the accuracy of the final optimum. Second, in some cases you do not want the optimizer to take large steps either. In case the cost function is suspected to possess large variations in a relatively small vicinity of the design space, large steps may result in too many trial steps, which do not improve the cost value. In these cases, it is safer to proceed with limited size steps and have more frequent improvements.

For these two limitations, the optimizer uses two independent distance measures. Both are based on user-defined quantities: the minimum and maximum step limits for individual optimization variables. Since the particular step is in a general direction, these measures are combined together in order to derive the limitation for that particular direction.

The step vector between the i^{th} and $(i+1)^{th}$ iterate is as follows:

$$s_i = x_{i+1} - x_i$$

The natural distance measure is,

$$\|s_i\| = \sqrt{s_i^T s_i}$$

which is the Euclidean norm.

A more general distance measure incorporates some "stretching" of the design space:,

$$\|s_i\|_D = \sqrt{s_i^T D^T D s_i}$$

where the matrix *D* incorporates the linear operation of the stretching of design space. The simplest case is when the *D* matrix is diagonal, meaning that the design space is stretched along the orthogonal direction of the base vectors.

The optimizer stops the search if,

$$\|s_i\|_{D_{min}} < 1$$

where D_{min} consists of diagonal elements

equal to the inverse of the **Min. Step** value assigned to the corresponding optimization variable. Similarly the optimizer truncates steps for which

$$\|s_i\|_{D_{max}} > 1$$

where D_{max} has diagonal elements equal to the inverse of **Max. Step** values of the corresponding optimization variables.

Related Topics

Setting the Min. and Max. Step Sizes

Cost Function

Adding a Cost Function

Setting the Min. and Max. Step Sizes

For the Quasi Newton and Pattern Search optimizers, the step size is the difference in a variable's value between one solved design variation and the next. The step size is determined when Optimetrics locates the next design variation that should be solved in an effort to meet the cost function.

1. In the **Setup Optimization** dialog box, click the **Variables** tab.
2. Optimetrics displays **Min Step** and **Max Step** columns, with default values for each variable to be optimized.
3. In the **Min Step** text box, type the minimum step size value. Optionally, modify the unit system in the **Units** text box.
4. In the **Max Step** text box, type the maximum step size value. Optionally, modify the unit

system in the **Units** text box.

5. Click **OK**.

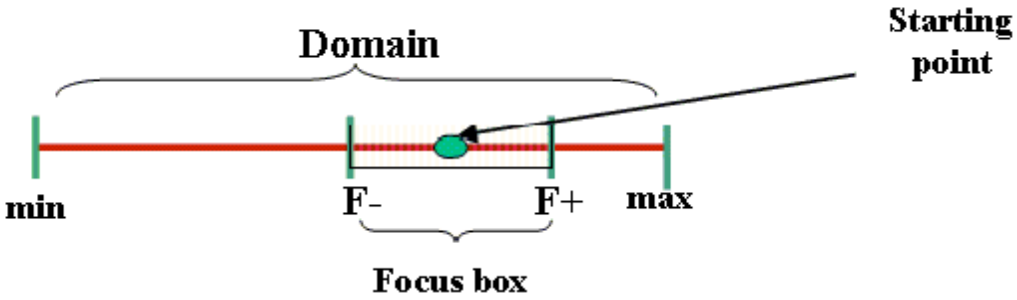
Hint	A value of zero is recommended for the minimum step size.

Related Topics

Step Size

Setting the Min and Max Focus

For the SNLP, SMINLP and Genetic Algorithm optimizers, the min focus and max focus criteria allow you to specify a sub-range of parameter values where the optimizer should look when performing the optimization. This focus box is where you suspect the optimal solution will be, so it is a hint for the optimizer.



- The domain limits the search. The domain = physical limits.
- The focus box does not limit the search. Rather, the Focus box = an initial guess of optimum search domain. The starting point is the center of the focus box, but the search does extend beyond the box.
- This focus must be inside the domain limits. Consequently, it has to be equal or smaller size. An error message is generated if you specify a focus outside the domain.
- The focus box must be at least one hundredth of the domain size. Otherwise, an error message is sent.

Equalizing the influence of different optimization variables.

The optimizer seeks optimal values for the optimization variables. These variables are usually quantities with specified units. The change in one variable could be measured in [mm] and the change in other variable could be measured in [mA]. Instead of those units, the optimizer uses internal abstract units, so that a change in one variable changes the design behavior about as much as the same change in another variable, where changes are measured in the respective internal abstract units. When you define the focus box, the unit of the abstract internal unit is

defined as the difference of the upper and lower focus limits. This way you can use the focus box to equalize the influence of different optimization variables on the design behavior.

To set the Min and Max Focus values:

1. In the **Setup Optimization** dialog box, click the **Variables** tab.
2. Optimetrics displays **Min. Focus** and **Max. Focus** columns, with default values for each variable to be optimized.

   If you do not have an initial guess based on your knowledge of the problem, make the focus box equal to the domain; that is, the physical limits. This tells SNLP to search the entire decision space.
   - In the **Min. Focus** text box, type the minimum value of the focus range. Optionally, modify the unit system in the **Units** text box.
   - In the **Max. Focus** text box, type the maximum value of the focus range. Optionally, modify the unit system in the **Units** text box.
   - Click **OK**.

Solving a Parametric Setup Before an Optimization

Solving a parametric setup before an optimization setup is useful for guiding Optimetrics during an optimization.

To solve a parametric setup before an optimization setup:

1. In the **Setup Optimization** dialog box, click the **General** tab.
2. In the **Parametric Analysis** pull-down list, click the parametric setup you want Optimetrics to solve before optimization.

Note	The parametric setup must include sweep definitions for the variables you are optimizing.

3. Select **Solve the parametric sweep before optimization**.

If the parametric setup has not yet been solved, Optimetrics solves it. Optimetrics uses the cost value evaluated at each parametric design variation to determine the next step in the optimization analysis. This enables you to guide the direction in which the optimizer searches for the optimal design variation.

Related Topics

Solving a Parametric Setup During an Optimization

Solving a Parametric Setup During an Optimization

Solving a parametric setup during an optimization analysis is useful when you want Optimetrics to solve every design variation specified in the parametric setup at each optimization iteration. A cost function goal could then depend on the value of the variable swept in the parametric setup.

To solve a parametric setup during an optimization analysis:

1. In the **Setup Optimization** dialog box, click the **General** tab.
2. In the **Parametric Analysis** pull-down list, click the parametric setup you want Optimetrics to solve during an optimization.
3. Select **Solve the parametric sweep during optimization**.
4. Optionally, you can adjust the sweep values to be used during the optimization.
   a. Click on the **Goal** tab, click **Setup Calculations** to specify a calculation.
      The **Add/Edit Calculation** dialog box is displayed.
   b. Click the **Calculation Range** tab.
   c. Click the **Edit** button for the sweep to be modified.
   d. In the pop-up dialog box, select the sweep values to use.
   e. Close the pup-up dialog box. Click **Done** to close the **Add/Edit Calculation** dialog.

Automatically Updating a Variable's Value After Optimization

When Optimetrics finds an optimal variable value by solving an optimization setup, it can automatically update that variable's current value set for the nominal model to the optimal value.

1. In the **Setup Optimization** dialog box, click the **General** tab.
2. Select **Update design parameters' values after optimization**.
   When optimization is complete, the current variable value for each optimized variable is changed to the optimal value.

Changing the Cost Function Norm

You can select the norm to be used in the calculation of the cost goal.

1. In the **Setup Optimization** dialog box, click the **Goals** tab.
2. Select **Show Advanced Options**.
3. Select a norm from the pull-down in the **Cost Function Norm Type** field. The options are **L1**, **L2**, and **Maximum**. **L2** is the default.

Related Topics

Explanation of L1, L2 and Max Norms in Optimization
Cost Function

Explanation of L1, L2 and Max norms in Optimization

When you set multiple goals for an optimization, the question arises as to what is actually going to drive the optimizer which is not a multi-objective one. The cost function will have a lot to do with it. The following discussion explains how the cost function is put together when there are multiple goals.

The general goal setting structure in Optimetrics is a logical sentence with the format:

Calculation$_{(i)}$ *Condition*$_{(i)}$ *Goal*$_{(i)}$ *Weight*$_{(i)}$

The cost function that the optimizer uses is built based on the norm setting as long as there are multiple goals and none of those use the “minimize” or “maximize” conditions. Thus, in this case the error associated with each individual goal (weighted) is combined in a way that is specific for each norm type chosen.

For **L1** norm the actual cost function uses the sum of absolute weighted values of the individual goal errors:

$$Cost = \sum_{1}^{N} \left| w_i \varepsilon_i \right|$$

For **L2** norm the actual cost function uses the weighted sum of absolute values of the individual

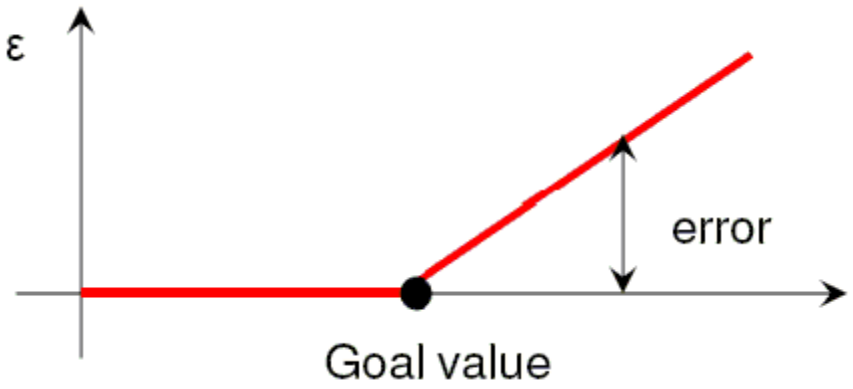
$$Cost = \sum_{1}^{N} w_i \varepsilon_i^2$$

For the **Maximum** norm the cost function uses the maximum among all the weighted goal errors:

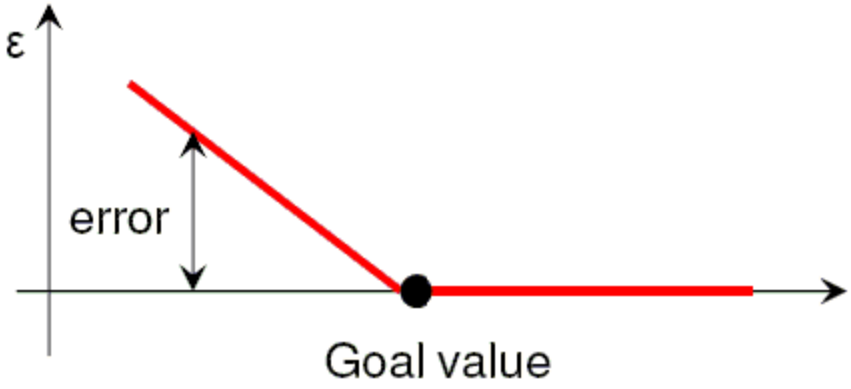
$$Cost = \max_{1}^{N} W_i \cdot \varepsilon_i$$

For all the above situations N is the number of individual goals $w_i\varepsilon_i$ are individual weighting factors and residual error respectively. A minimization of the cost function is performed during optimization since it makes sense to minimize the error in the sense of the chosen norm type.

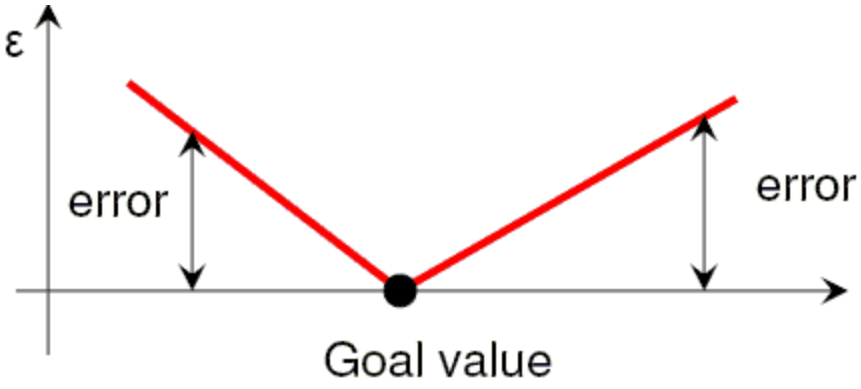
The graphical representation of the error is possible and depends upon the actual condition being used. If a “<” condition is used, the error can be represented as below:



If a “>” condition is used, the error can be represented as below:



If a “=” condition is used, the error is double-sided and can be represented as below:



The norm type doesn’t impact goal setting that use as condition the “minimize” or “maximize” scenarios. Note that when using “minimize” or “maximize” settings for the condition there should be a single goal setting which in this case coincides with the cost function.

Related Topics

Cost Function

Advanced Genetic Algorithm Optimizer Options

The Genetic Algorithm (GA) search for Optimization analysis is an iterative process that goes through a number of generations. In each generation some new individuals (Children / Number of Individuals) are created and the so grown population participates in a selection (natural-selection) process that in turn reduces the size of the population to a desired level (Next Generation / Number of Individuals).

If you select the Genetic Algorithm for an Optimization analysis, a **Setup** button is enabled on the **Setup Optimization** page.

1. Click the **Setup** button to open the **Advanced Genetic Algorithm Optimizer Options** dialog.
2. Select the Stopping Criteria. Any of the three following, or any combination of these can be selected.
   - **Maximum number of generations**. If checked, this enables a value field.
   - **Elapsed time**. If checked, this enables a drop down menu with times ranging from five minutes to two weeks.
   - **Slow convergence**.
3. Specify the Parents.

   The first step toward mating is a selection process that determines the participating individuals. Potential parents are selected from the Current Generation. This is a set of individuals that is always a subset of the current generation.
   - **Number of individuals** value field -- specify the number of parents for the optimizer to use. You can set the Number of Individuals to less than or equal to the size of the "Current Generation". One reason to consider fewer parents than the possible maximum is to steer the GA toward improvement by selecting the better portion of the current generation to be able to mate.
   - **Roulette selection** check box -- if checked, this enables the **Selection pressure** value field. This number defines how many times more probable is the selection of the best individual over the worst individual in an elementary spin of the roulette wheel.
4. Specify the Mating pool.

   The Mating pool is created by selecting randomly from the parents, but with each selection, the parent gets "cloned" so it can be selected again and again.
   - **Number of individuals** field -- specify the number individuals to include in the mating pool.
   - **Reproduction setup**-- this button opens the **Genetic Algorithm Optimizer Reproduction Setup** dialog box.

5. Click the **Reproduction setup** button for the dialog to specify the Crossover setup, and the Mutation setup.

   The crossover and mutation operator have different roles: *Crossover* mixes "features" of the parents in a new combination, while *mutation* slightly alters the "features" of the individuals. Both need to be present in a GA. The crossover is a way to discover new combinations while the mutation acts as a local search or fine-tuning step. Mutation also keeps diversity in a population, which is a must for GA.

   The crossover operator has two steps. It first alters the variable values of the parents according to a distribution. This tends to produce one child that looks a lot like one parent, and one child that looks a lot like the other parent. Next, some of the variable values of the two children can be exchanged in order to achieve more variation.

   For crossover there are four possible parameters.

   a. **Individual Crossover Probability** determines, for each pair in the mating pool, the probability that their features will be mixed. Usually, this probability should be close or equal to one. If you set it set less than one, some parents will produce two children which are exact clones of the parents. This means that some children inherit all the features of their parents unchanged.
   b. Parents often have multiple variables. If the parent is a candidate for mixing, the **Variable Crossover Probability** determines, for each variable, the probability of mixing. This is usually set high to ensure that most or all variables mix.
   c. **Variable Exchange Probability**: After the slight change in the variable values has been made, the crossover operation is also able to exchange the values of the variables between the two children that are being constructed. The Variable Exchange Probability governs the likelihood of exchange of any variable.
   d. **Mu** is a general parameter defining the sharpness of the distribution that might be used for the **Variable Crossover Probability**. Mu should be greater than one. There is no theoretical upper limit, but we recommend not exceeding 30.

6. Select one of the four **Crossover types** from the drop-down menu.

   The crossover type selected affects the options available.

Uniform	Individual crossover probability Variable crossover probability
One point	Individual crossover probability
Two point	Individual crossover probability
Simulated binary crossover	Individual crossover probability Variable crossover probability

	Variable exchange probability Mu

7. Select the **Mutation type**--this can be one of three types, which you select from a drop-down menu.
   - **Uniform Distribution**
   - **Gaussian Distribution**
   - **Polynomial Mutation.**
8. For the selected mutation type, set the following parameters:
   - **Uniform Mutation Probability**: If this is more than zero (recommendation is to have still a small probability here), then there will be some children whose features are simply a completely random design (design variables randomly selected over the domain).
   - **Individual Mutation Probability** controls, for each child, the likelihood of a mild mutation.
   - **Variable Mutation Probability**. If the child will be mutated, this probability controls at the variable level the likelihood of a mutation of the variables.
   - **Standard Deviation** is the standard deviation of the selected distribution that is being used for the mutation and it is measured relatively to the optimization-domain.
9. When you have completed the Reproduction setup in the **Genetic Algorithm Optimizer Reproduction Setup** dialog, click **OK** to close it and return to the **Advanced Genetic Algorithm Optimizer Options** dialog.
10. In the **Advanced Genetic Algorithm Optimizer Options** dialog box, specify the children as a Number of Individuals.
11. Set the **Pareto Front** value.

    This the number of the very best individuals (identified relative to the cost function) to keep for future generations.
12. Set the Next Generation parameters. The Next Generation is selected from the Parents, the children, and the Pareto front.
    - **Number of individuals** value field -- specify the number of individuals to survive to form the next generation for the optimizer to use.
    - **Roulette selection** check box -- if checked, this enables the **Selection pressure** value field. This number defines how many times more probable is the selection of the best individual over the worst individual in an elementary spin of the roulette wheel.
13. Click **OK** to accept the settings for the Genetic Algorithm and to close the dialog.

Related Topics

Setting up an Optimization Analysis

Adding a cost function

Optimization Overview

Acceptable Cost

Explanation of L1, L2, and Max Norms in Optimization
Choosing an Optimizer

Sensitivity Analysis Overview

During a sensitivity analysis, Optimetrics explores the vicinity of the design point to determine the sensitivity of the design to small changes in variables. The variables and their attributes define the design point, the problem around which the sensitivity analysis is performed.

When Optimetrics performs a sensitivity analysis, its goal is to calculate the second-order regression polynomials for all of the design's output parameters. The algorithm first determines an appropriate interval for each variable. The intervals are further sub-divided according to the available number of iterations and variables. If the primary output is not used, the specified initial displacement values define those intervals.

When all of the design calculations are complete, the second-order polynomials are fitted for all the output parameters. Optimetrics then reports the following quantities:

- Regression value at the current variable value.
- First derivative of the regression.
- Second derivative of the regression.

Related Topics

Setting Up a Sensitivity Analysis
Selecting a Primary Output

Selecting a Primary Output

During a sensitivity analysis, the design variations that Optimetrics selects to solve are close to the design point, but not so close that numerical noise (from the finite element mesh) affects the analysis. The algorithm that Optimetrics uses to determine the design variations to solve must be based on only one output parameter and that output parameter's numerical noise. Therefore, if you have defined more than one output parameter, be sure to select **Primary Output** for the output variable on which you want the selection of design variations to be based.

Related Topics

Setting Up an Output Parameter
Setting Up a Sensitivity Analysis

Setting Up a Sensitivity Analysis

Following is the general procedure for setting up a sensitivity analysis. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the

whole process for minor changes. You can create a sensitivity setup before defining variables but all variables must be defined before you start the sensitivity analysis.

1. Before a variable can be included in a sensitivity analysis, you must specify that you intend for it to be used during a sensitivity analysis in the **DesignProperties** dialog box.
2. On the **Maxwell3D or Maxwell2D** menu, point to **Optimetrics Analysis**, and then click **Add Sensitivity**  .

   The **Setup Sensitivity Analysis** dialog box appears.
3. Under the **Calculations** tab, type the maximum number of iterations per variable value that you want Maxwell to perform in the **Max. No. of Iterations/Sensitivity Variable** text box.
4. Set up an output parameter calculation and select a Primary Output
5. Specify the value of the design point at which the sensitivity analysis should stop in the **Approximate Error in Primary Output** text box.
6. In the **Variables** tab, specify the **Min/Max** values for variables included in the optimization, and the **Initial Displacement (Initial Disp.)** for the analysis.

   You may also override the variable starting values by clicking the **Override** check box and entering the desired value in the **Starting Value** field.
7. In the **General** tab, specify whether Optimetrics should use the results of a previous Parametric analysis or perform one as part of the optimization process.
8. Under the **Options** tab, if you want to save the field solution data for every solved design variations in the optimization analysis, select **Save Fields And Mesh**.

Note	Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.

   You may also select **Copy geometrically equivalent meshes** to reuse the mesh when geometry changes are not required, for example when optimizing on a material property or source excitation.This will provide some speed improvement in the overall optimization process.

The following ***optional*** sensitivity analysis setup options can also be used:

- Modify the starting variable value.
- Modify the minimum and maximum values of variables that will be solved.
- Exclude variables from the sensitivity analysis.
- Set the initial displacement.
- Modify the values of fixed variables that are not being modified during the sensitivity analysis.
- Set linear constraints.
- Request that Optimetrics solve a parametric sweep **before** a sensitivity analysis.
- You can also request that Optimetrics solve a parametric sweep **during** a sensitivity analysis.

- Set **HPC and Analysis Options**, which allows you to select or create an analysis configuration.

Note	Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

Related Topics

Sensitivity Analysis Overview

Setting the Maximum Iteration Per Variable

Setting the Maximum Iterations Per Variable

The **Max. No. of Iterations/Sensitivity Variable** value is the maximum number of design variations that Optimetrics solves per variable during a sensitivity analysis. This value is a stopping criterion; if the maximum number of iterations has been completed, the sensitivity analysis stops. If the maximum number of iterations has not been completed, the sensitivity analysis continues by performing another iteration, that is, by solving another design variation. It performs iterations until the approximate error in primary output value is reached or until Optimetrics cannot proceed as a result of other sensitivity setup constraints, such as when it searches for a variable value that is larger than the maximum value.

To set the maximum number of iterations for a sensitivity analysis:

- Under the **Calculations** tab of the **Setup Sensitivity Analysis** dialog box, type a value in the **Max. No. of Iterations/Sensitivity Variable** text box.

Related Topics

Setting Up an Output Parameter

Setting Up an Output Parameter

Following is the general procedure for adding an output parameter to a sensitivity setup:

1. Under the **Calculations** tab of the **Setup Sensitivity Analysis** dialog box, click **Setup Calculations** to open the **Add/Edit Calculations** dialog box.
2. In the **Add/Edit Calculations** dialog box, set up output parameter calculations to be evaluated for sensitivity.
3. To modify the solution from which the output parameter is to be extracted, click in the **Solution** column and select from the options in the pop-up list.
4. You can modify the Calculation specified by clicking on the output parameter in the table and selecting **Edit**.
5. For output parameters based on swept variable, you must choose a single value in the **Calculation Range** at which to evaluate the output parameter.

6. If you have more than one output parameter, select **Primary Output** if you want Optimetrics to use the output parameter to base its selection of solved design variations.

Note	During a sensitivity analysis, the design variations that Optimetrics selects to solve are close to the design point, but not so close that numerical noise (from the finite element mesh) affects the analysis. The algorithm that Optimetrics uses to determine the design variations to solve must be based on only one output parameter and that output parameter's numerical noise. If you have defined more than one output parameter, be sure to select **Primary Output** for the output variable on which you want the selection of design variations to be based.

Related Topics

Selecting a Primary Output

Specifying a Solution Quantity for an Output Parameter

When setting up an output parameter, you must identify the solution quantity on which to base the output parameter. Solution quantities are specified by mathematical expressions that are composed of basic quantities, such as matrix parameters; and output variables.

The **Add/Edit Calculation** dialog box allows you to define the mathematical equation for one or multiple output parameters. To set up an output parameter:

1. In the **Context** section of the dialog:
   - Select the **Report Type** with a pull-down selection list containing the available types for this design.
   - Select the **Solution** from the drop down selection list. This lists the available setups and sweeps. As a minimum, the **LastAdaptive** solution is available.
   - Select the **Geometry** from the drop down selection list or select none (the default). This modifies the list of quantities available to the ones that apply to the specific geometry.
   - When selecting a geometry, you may also be required to specify a point within the geometry where the calculation is to be performed.
2. The **Output Variables** button opens the Output Variables dialog box allowing you to create special output variables to be used in the output parameter.
3. The **Calculation Expression** field in the **Trace** tab is used to enter the equation to be used for the output parameter. To enter an expression, you may type it directly into the field or use the **Category**, **Quantity**, and **Function** lists as follows:
   - Select the **Category**, these depend on the Solution type and the design. This lets you specify the category of information to be used in the output parameter.
   - Select a **Quantity** from the list. Available quantities depend upon the Solution type, as well as the Geometry and Category selection. Selecting a Quantity automatically enters it into the Calculation Expression field.
   - Select a **Function** to apply to the value in the calculated expression.

- For swept variables, the Range **Function** button opens the **Set Range Function** dialog to apply functions to the expression that apply over the sweep range.

4. The **Calculation Range** tab applies to swept variables and allows you to specify the range of the sweep over which to apply the calculation.
5. When the desired **Calculation Expression** has been obtained, click the **Add Calculation** button to add the entry to the calculation table in the Setup Sensitivity Analysis dialog box. You may add multiple entries to the table simply by changing the **Calculation Expression** and using the **Add Calculation** button.
6. To update or edit a selected cost function, enter the desired Calculation Expression and click the **Update Calculation** button.
7. Click **Done** to return to the Setup Sensitivity Analysis dialog box.

Note	The solution quantity you specify must be able to be evaluated to a single, real number.

Related Topics

Setting the Calculation Range of an Output Parameter

Setting the Calculation Range of an Output Parameter

The calculation range of a solution quantity determines the intrinsic variable value at which the solution quantity is to be extracted. For a sensitivity setup, the calculation range must be a single value. If you specified that the solution quantity be extracted from a frequency sweep solution, by default, Optimetrics uses the starting frequency in the sweep.

1. Under the **Calculations** tab of the **Setup Sensitivity Analysis** dialog box, click **in the Calculation Range column of the table for the calculation to be modified.**

   The **Edit Calculation Range** dialog box appears.
2. In the **table**, click the **Edit** button in the row to be modified.

   If you choose to solve a parametric setup during the sensitivity analysis, the variables swept in that parametric setup are available in the pop-up list dialog box. If you sweep a variable in the parametric setup that is also a sensitivity variable, that variable is excluded from the sensitivity analysis.
3. **Click on the value for the calculation range in the list and dismiss the pop-up dialog box**.
4. Click **OK in the Edit Calculation Range dialog box to accept the new value for the intrinsic variable, and return to the Setup Sensitivity Analysis dialog box**.

Related Topics

Setting Up an Output Parameter

Modifying the Starting Variable Value for Sensitivity Analysis

The design point of the sensitivity analysis is the starting value of the sensitivity variable and is usually the first variation to be solved. Optimetrics automatically sets the starting value of a variable to be the current value set for the nominal design. You can modify the design point for each sensitivity setup.

Warning	Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

1. In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.

   All of the variables that were selected for the sensitivity analysis are listed.
2. Type a new value in the **Starting Value** text box for the value you want to override, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered is to be used for this sensitivity analysis; the current value set for the nominal model will be ignored.
   - Alternatively, you can select the **Override** option first, and then type a new variable value in the **Starting Value** text box.
3. Optionally, click a new unit system in one of the **Units** text boxes.

To revert to the default starting value, clear the **Override** option.

Related Topics

Setting Up a Sensitivity Analysis

Setting the Min. and Max. Variable Values

For every sensitivity setup, Optimetrics automatically sets the minimum and maximum values that it will consider for a sensitivity variable. Optimetrics sets a variable's minimum value equal to approximately one-half its starting value. (The starting value is the variable's current value set for the nominal design.) Optimetrics sets the variable's maximum value equal to approximately 1.5 times the starting value. During sensitivity analysis, variable values outside this range are not considered.

Warning	Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

Related Topics

Override the default minimum and maximum variable values for a single sensitivity setup.

Change the default minimum and maximum variable values for every sensitivity setup.

Overriding the Min. and Max. Variable Values for a Single Sensitivity Setup

1. In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.

   All of the variables that were selected for sensitivity analysis are listed.
2. Type a new value in the **Min** or **Max** text box for the value you want to override, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered is to be used for this sensitivity analysis; the variable's current **Min** or **Max** value set in the nominal design is ignored.
   - Alternatively, you can select the **Override** option first, and then type a new value in the **Min** or **Max** text box.
3. Optionally, click a new unit system in one of the **Units** text boxes.

To revert to the default minimum and maximum values, clear the **Override** option.

Related Topics

Setting Up a Sensitivity Analysis

Changing the Min. and Max. Variable Values for Every Sensitivity Setup

1. Make sure the variable's minimum and maximum values are not being overridden in any sensitivity setup.
2. If the variable is a design variable, do the following: Click **Maxwell3D or Maxwell2D>Design Properties**.

   If the variable is a project variable, do the following: Click **Project**>**Project Variables**.

   The **Properties** dialog box appears.
3. Select **Sensitivity**.
4. Type a new value in the **Min** or **Max** text box for the value you want to override, and then press **Enter**.

   When Optimetrics solves a sensitivity setup, it does not consider variable values that lie outside of this range.

Related Topics

Setting Up a Sensitivity Analysis

Setting the Initial Displacement

The initial displacement is the difference in a variable's starting value and the next solved design variation. During the sensitivity analysis, Optimetrics does not consider an initial variable value that is greater than this step size away from the starting variable value.

1. In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.
2. Optimetrics displays the **Initial Disp.** column, with default values for each sensitivity variable.
3. In the **Initial Disp.** text box, type the initial displacement value. Optionally, modify the unit system in the **Units** text box.

Related Topics

Setting Up a Sensitivity Analysis

Solving a Parametric Setup Before a Sensitivity Analysis

Solving a parametric setup before a sensitivity setup is useful for guiding Optimetrics in a sensitivity analysis.

To solve a parametric setup before a sensitivity setup:

1. In the **Setup Sensitivity Analysis** dialog box, click the **General** tab.
2. Click the parametric setup you want Optimetrics to solve before the sensitivity setup from the **Parametric Analysis** pull-down list.

Note	The parametric setup must include sweep definitions for the sensitivity variables.

3. Select **Solve the parametric sweep before analysis**.

If the parametric setup has not yet been solved, Optimetrics solves it. Optimetrics uses the results (of the solution calculation you requested under the **Goals** tab of the **Setup Sensitivity** dialog box) to determine the next design variation to solve for the sensitivity analysis.

Related Topics

Setting Up a Sensitivity Analysis

Solving a Parametric Setup During a Sensitivity Analysis

Solving a parametric setup during a sensitivity analysis is useful when you want Optimetrics to solve every design variation in the parametric setup at each sensitivity analysis iteration. An output parameter goal could then depend on the value of the variable swept in the parametric setup.

To solve a parametric setup during a sensitivity analysis:

1. In the **Setup Sensitivity Analysis** dialog box, click the **General** tab.
2. Click the parametric setup you want Optimetrics to solve during the sensitivity analysis from the **Parametric Analysis** pull-down list.
3. Select **Solve the parametric sweep during analysis**.

Related Topics

Setting Up a Sensitivity Analysis

Statistical Analysis Overview

Statistical analysis allows you to explore the effects of random combinations of values of selected variables on selected global or local available analysis results. Therefore, before a variable can be included in a statistical analysis, you must specify that you intend for it to be used during a statistical analysis. For each variable you must specify the type of distribution (Uniform, Gaussian, Lognormal or User Defined) and the corresponding parameters of the selected distribution.

Note	A statistical analysis is currently limited to using a maximum of 30 variables.

In addition to specifying the variables to be used in the statistical analysis and the parameters of the chosen distribution, the output quantities of interest also need to be specified. These quantities can be global ones such as previously defined parameters (Force/torque, inductance / capacitance, etc), other named quantities, quantities defined in the field calculator as global (such a domain integral of a certain field quantity) or local (such as field value at a certain location). The calculations to be performed during the statistical analysis are specified during setup, in a manner similar to other types of analysis in Optimetrics.

Following the analysis the statistical distribution of the output quantities can be visualized in histogram format. To access available reports, after the statistical analysis is complete, right click the respective Statistical analysis setup and select **View Analysis Result**.

Related Topics

Setting Up a Statistical Analysis

Setting Up a Statistical Analysis

Following is the general procedure for setting up a statistical analysis. Once you have created a setup, you can **Copy** and **Paste** it, and then make changes to the copy, rather than redoing the whole process for minor changes. You can create a statistical setup before defining variables but all variables must be defined before you start the statistical analysis.

1. Before a variable can be included in a statistical analysis, you must specify that you intend for it to be used during a statistical analysis in the **Properties** dialog box.
2. On the **Maxwell3D or Maxwell2D menu, point to Optimetrics Analysis,** and then click **Add Statistical**  . The **Setup Statistical Analysis** dialog box appears.

3. Under the **Calculations** tab, type the maximum number of iterations you want Maxwell to perform in the **Maximum Iterations** text box.
4. If you want to specify an initial seed value, select the **Specify initial seed value** check box and enter a positive value in the text box. Each different seed value creates a new statistical sequence.
5. Specify a solution quantity to evaluate.
6. In the **Calculation** text box, set the value at which the solution quantity is to be computed.
7. Optionally, modify the distribution criteria to be used.
8. The following ***optional*** statistical analysis setup options can also be used:
   - Modify the starting variable value.
   - Exclude variables from the statistical analysis.
   - Modify the values of fixed variables that are not being modified during the statistical analysis.
   - Request that Optimetrics solve a parametric sweep during a statistical analysis.

Note	Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

   - Set **HPC and Analysis Options**, which allows you to select or create an analysis configuration.
9. If you want to save the field solution data for the design variations solved during analysis, select **Save Fields**.

Related Topics

Statistical Analysis Overview

Setting the Maximum Iterations for a Statistical Analysis

The **Maximum Iterations** value is the maximum number of design variations Optimetrics solves during a statistical analysis. This value is a stopping criterion; if the maximum number of iterations has been completed, the analysis stops. If the maximum number of iterations has not been completed, Optimetrics continues by performing another iteration, that is, by solving another design variation.

To set the maximum number of iterations for a statistical analysis:

- Under the **Calculations** tab of the **Setup Sensitivity Analysis** dialog box, type a value in the **Maximum Iterations** text box.

Related Topics

Setting up a Statistical Analysis

Specifying an Initial Seed Value for a Statistical Analysis

The Initial Seed Value is a positive integer used as the starting point during a statistical analysis. When you enter a new seed value, a new statistical sequence is created.

To set an initial seed value for a statistical analysis:

- Under the **Calculations** tab of the **Setup Sensitivity Analysis** dialog box, select the **Specify initial seed value** check box and type a positive value in the text box.

Related Topics

Setting up a Statistical Analysis

Specifying the Solution Quantity to Evaluate for Statistical Analysis

When you add a statistical setup, you can identify one or more solution quantities to evaluate. The solution quantities are specified by mathematical expressions that are composed of basic quantities. When you view the results, Maxwell displays the distribution of the solution quantities.

1. In the **Calculations** tab of the **Setup Statistical Analysis** dialog box, click **SetupCalculations**.

   The **Add/Edit Calculations** dialog box is displayed, allowing you to define one or more mathematical expressions for statistical evaluation.
2. In the **Context** section of the dialog:
   - Select the **Report Type** with a pull-down selection list containing the available types for this design.
   - Select the **Solution** from the drop down selection list. This lists the available setups and sweeps. As a minimum, the **LastAdaptive** solution is available.
   - Select the **Geometry** from the drop down selection list or select none (the default). This modifies the list of quantities available to the ones that apply to the specific geometry.
   - When selecting a geometry, you may also be required to specify a point within the geometry where the calculation is to be performed.
3. The **Output Variables** button opens the Output Variables dialog box allowing you to create special output variables to be used in the output parameter.
4. The **Calculation Expression** field in the **Trace** tab is used to enter the equation to be used for the solution quantities. To enter an expression, you may type it directly into the field or use the **Category**, **Quantity**, and **Function** lists as follows:
   - Select the **Category**, these depend on the Solution type and the design. This lets you specify the category of information to be used in the output parameter.
   - Select a **Quantity** from the list. Available quantities depend upon the Solution type, as well as the Geometry and Category selection. Selecting a Quantity automatically enters it into the Calculation Expression field.
   - Select a **Function** to apply to the value in the calculated expression.
   - For swept variables, the Range **Function** button opens the **Set Range Function** dialog to apply functions to the expression that apply over the sweep range.

5. The **Calculation Range** tab applies to swept variables and allows you to specify the range of the sweep over which to apply the calculation.
6. When the desired **Calculation Expression** has been obtained, click the **Add Calculation** button to add the entry to the calculation table in the Setup Statistical Analysis dialog box. You may add multiple entries to the table simply by changing the **Calculated Expression** and using the **Add Calculation** button.
7. To update or edit a selected cost function, enter the desired Calculation Expression and click the **Update Calculation** button.
8. Click **Done** to return to the Setup Statistical Analysis dialog box.

Note	The solution quantity you specify must be able to be evaluated to a single, real number.

Related Topics

Setting up a Statistical Analysis

Setting the Maximum Iterations for a Statistical Analysis

Setting the Solution Quantity's Calculation Range

The calculation range of a solution quantity determines the intrinsic variable value at which the solution quantity is extracted. For a statistical setup, the calculation range must be a single value. If you specified that the solution quantity be extracted from a frequency sweep solution, Optimetrics will use the starting frequency in the sweep by default.The calculation range should be set during the setup of the solution quantity for statistical evaluation. In order to modify the calculation range, do the following:

1. Under the **Calculations** tab of the **Setup Statistical Analysis** dialog box, click **in the Calculation Range column of the table for the calculation to be modified**.

   The **Edit Calculation Range** dialog box appears.
2. In the **table**, click the **Edit** button in the row to be modified.

   If you choose to solve a parametric setup during the statistical analysis, the variables swept in that parametric setup are available in the pop-up list dialog box. If you sweep a variable in the parametric setup that is also a statistics variable, that variable is excluded from the statistics analysis.
3. **Click on the value for the calculation range in the list and dismiss the pop-up dialog box**.
4. Click **OK in the Edit Calculation Range dialog box to accept the new value for the intrinsic variable, and return to the Setup Statistical Analysis dialog box**.

Related Topics

Setting up a Statistical Analysis

Setting the Distribution Criteria

For every statistical setup, Optimetrics automatically sets the distribution criteria to be uniform within a 10% tolerance of the variable's starting value. You can modify the distribution type and criteria for a single statistical setup or for every statistical setup.

Related Topics

Override the default distribution criteria for a single statistical setup.

Change the default distribution criteria for every statistical setup.

Overriding the Distribution Criteria for a Single Statistical Setup

To override the default distribution criteria for a single statistical setup:

1. In the **Setup Statistical Analysis** dialog box, click the **Variables** tab.

   All of the variables that were selected for statistical analysis are listed.
2. Check or clear the **Include** check box for each variable to define the specific variables to be varied in the statistical analysis setup.
3. For each included variable, select **Uniform**, **Gaussian**, **Lognormal**, or **User Defined** in the **Distribution** column for the variable you want to override.

   If you changed the distribution type, the **Override** option is now selected. This indicates that the distribution type you selected is to be used for this optimization analysis; the current distribution type selected for the variable in the nominal design is ignored in this statistical analysis.
   - Alternatively, you can select the **Override** option first, and then select a different distribution type in the **Distribution** text box.
4. Optionally, if you want to change the distribution criteria, click in **Distribution Criteria** column for the variable you want to override.

   The **Edit Distribution** dialog box appears.
5. If the distribution type is **Gaussian**, do the following:
   a. Type the lower limit of the distribution in the **Cutoff Probability** text box. This is a value =>0 and < 0.1.
   b. Type the mean value of the distribution in the **Mean** text box.
   c. Type the standard deviation of the distribution in the **Std Dev** text box.
   Maxwell will solve design variations using a Gaussian distribution within the low and high cutoff values.
6. If the distribution type is **Uniform**, do the following:
   - Enter a tolerance value in the text box.

     Maxwell will solve design variations within the tolerance range of the starting value, using an even distribution.
7. If the distribution type is **Lognormal**, do the following:

a. Enter the cutoff probability in the **Cutoff Probability** text box.
b. Enter the sigma value of the distribution in the **Sigma** text box and select a unit from the pull-down.
c. Enter the m value of the distribution in the **M** text box.
d. Enter the theta value in the Theta text box and select a unit from the pull-down.

8. If the distribution type is User Defined, do the following:
   a. Enter the cutoff probability in the **Cutoff Probability** text box.
   b. Click **Edit XY Data** to open the **Edit Datasets** dialog box in which you can select an existing dataset, or create a new one.
9. By default, all variables are set to sample using **Latin Hypercube** sampling. This sampling method provides for greater variability than random sampling by keeping track of chosen samples and guaranteeing that samples cannot be repeated. You may revert to random sampling by clearing the check box in the **Latin Hypercube** column for any desired variable.
10. Click **OK**.

To revert to the default distribution settings, clear the **Override** option.

Related Topics

Statistical Cutoffs

Changing the Distribution Criteria for Every Statistical Setup

To change the default distribution criteria for every statistical setup:

1. Make sure that the variable's distribution criteria are not being overridden in any statistical setup.
2. If the variable is a design variable, do the following: On the **Maxwell3D or Maxwell2D** menu, click **Design Properties**.

   If the variable is a project variable, do the following: Click **Project**>**Project Variables**.

   The **Properties** dialog box appears.
3. Select **Statistics**.
4. Click in the **Distribution** column for the variable you want to change, and then select **Uniform**, **Gaussian**, **Lognormal**, or **User Defined**.
5. Optionally, if you want to change the distribution criteria, click in the **Distribution Criteria** column for the variable you want to change.

   If the distribution type is **Gaussian**, the **Gaussian Distribution** dialog box appears. If the distribution type is **Uniform**, the **Uniform Distribution** dialog box appears.
6. If the distribution type is **Gaussian**, do the following:
   a. Type the lower limit of the distribution in the **Cutoff Probability** text box. This is a value =>0 and < 0.1.

b. Type the mean value of the distribution in the **Mean** text box.
c. Type the standard deviation of the distribution in the **Std Dev** text box.
Maxwell will solve design variations using a Gaussian distribution within the low and high cutoff values.

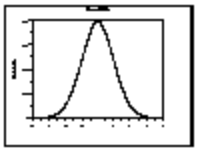
7. If the distribution type is **Uniform**, do the following:
   a. Type a cutoff probability value in the **Cutoff Probability** text box.
   b. Type mean and tolerance values in the corresponding text boxes.
   Maxwell will solve design variations within the tolerance range of the starting value, using an even distribution.
8. If the distribution type is **Lognormal**, do the following:
   a. Type a cutoff probability value in the **Cutoff Probability** text box.
   b. Type values for Sigma, M, and Theta in the corresponding text boxes.
9. If the distribution type is **User Defined**, do the following:
   a. Type a cutoff probability value in the **Cutoff Probability** text box.
   b. Click **Edit XY Data** to open the **Edit Dataset** dialog.
   c. Either type or import the X and Y data values for the distribution in the **Edit Dataset** dialog.
10. Click **OK**.

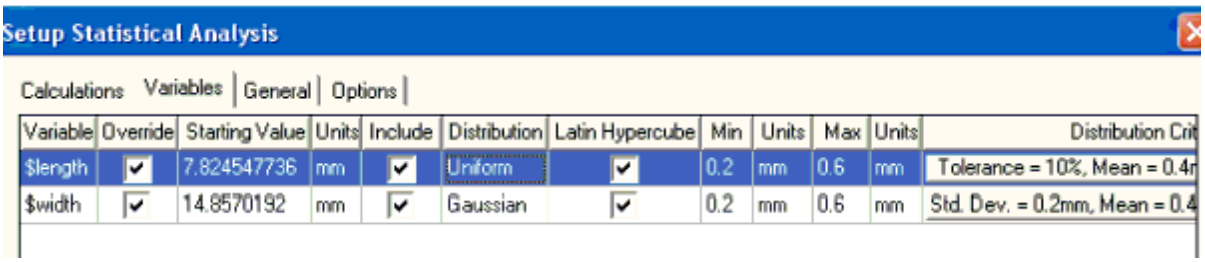
Related Topic

Statistical Cutoffs

Statistical Cutoffs

The cutoff probability values affects the Gaussian distribution criteria. This is a value =>0 and < 0.1. Maxwell solves design variations using a Gaussian distribution using a lower limit cutoff probability and specified mean and standard deviation values.



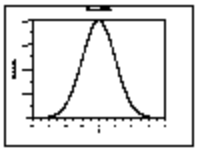


Uniform distributions such as variable “length” above use only the Tolerance value, and do not have a cutoff probability.

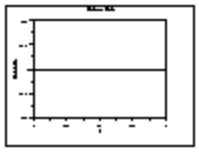
Edit Distribution

When setting the distribution type for a variable, you have the option of changing the distribution parameters from the default values.

1. If the distribution type is **Gaussian**, do the following:
   a. Type the lower limit of the distribution in the **Cutoff Probability** text box. This is a value =>0 and < 0.1.
   b. Type the mean value of the distribution in the **Mean** text box.
   c. Type the standard deviation of the distribution in the **Std Dev** text box.
   Maxwell solves design variations using a Gaussian distribution within the specified mean and standard deviation values.

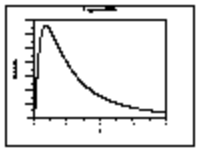


2. If the distribution type is **Uniform**, do the following:
   a. Type the lower limit of the distribution in the **Cutoff Probability** text box.
   b. Type the mean value of the distribution in the **Mean** text box.
   c. Enter the tolerance in the **Tolerance** text box.
   Maxwell solves design variations within the tolerance range of the starting value, using an even distribution.



3. If the distribution type is **Lognormal**, do the following:
   a. Type the lower limit of the distribution in the **Cutoff Probability** text box.
   b. Enter the shape parameter of the distribution in the **Sigma** text box.
   c. Enter the scale parameter in the **M** text box. The scale parameter should be set to 1 for the standard lognormal distribution.
   d. Enter the location parameter value for **Theta** in the text box. The value for a standard lognormal distribution is 0.

.Maxwell solves design variations with a logarithmic distribution using the shape, scale and location parameters provided.



4. If the distribution type is **User Defined**, do the following:
   a. Type the lower limit of the distribution in the **Cutoff Probability** text box.
   b. Select the **Edit XY Data** button to manually define the data distribution using datasets.

Related Topics

Adding Datasets
Changing the Distribution Criteria for Every Statistical Setup
Overriding the Distribution Criteria for a Single Statistical Setup

Modifying the Starting Variable Value for Statistical Analysis

A variable's starting value is the first value that is solved during the statistical analysis. Optimetrics automatically sets the starting value of a variable to be the current value set for the nominal design. You can modify this value for each statistical setup.

Warning	Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

1. In the **Setup Statistical Analysis** dialog box, click the **Variables** tab.

   All of the variables selected for the statistical analysis are listed.
2. Type a new value in the **Starting Value** text box for the value you want to override, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered is to be used for this statistical analysis; the current value set for the nominal model will be ignored.
   - Alternatively, you can select the **Override** option first, and then type a new variable value in the **Starting Value** text box.
3. Optionally, click a new unit system in one of the **Units** text boxes.

To revert to the default starting value, clear the **Override** option.

Related Topics

Setting up a Statistical Analysis

Solving a Parametric Setup During a Statistical Analysis

Solving a parametric setup during a statistical analysis is useful when you want Optimetrics to solve every design variation in the parametric setup at each statistical analysis iteration.

To solve a parametric setup during a statistical analysis:

1. In the **Setup Statistical Analysis** dialog box, click the **General** tab.
2. Click the parametric setup you want Optimetrics to solve during the statistical analysis from the **Parametric Analysis** pull-down list.
3. Select **Solve the parametric sweep during analysis**.

Related Topics

Setting up a Statistical Analysis

Using Design of Experiments

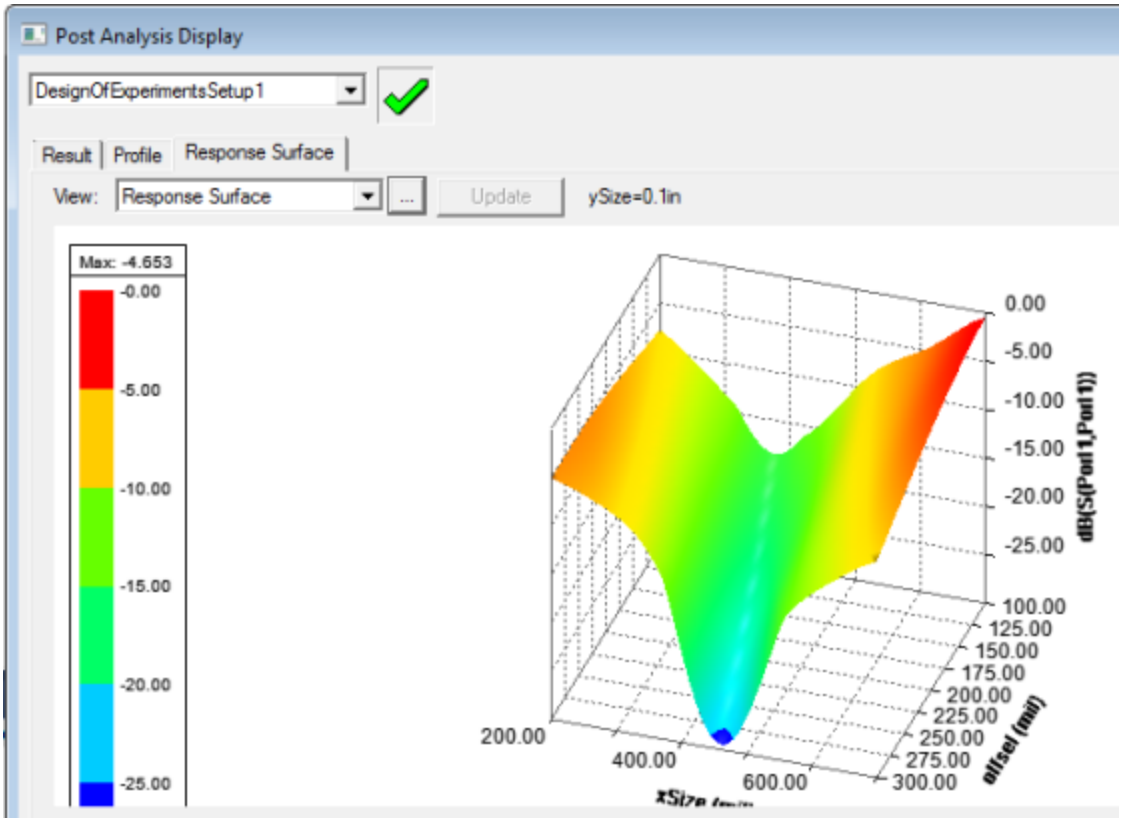
Design of Experiments (DOE) is a technique used to scientifically determine the location of sampling points and is included as part of the Response Surface, Goal Driven Optimization, and Analysis systems. Design of Experiments plus a mathematical approximation of output parameters lets you:

- Reduce the number of simulations
- Interactively explore the design space before running optimization

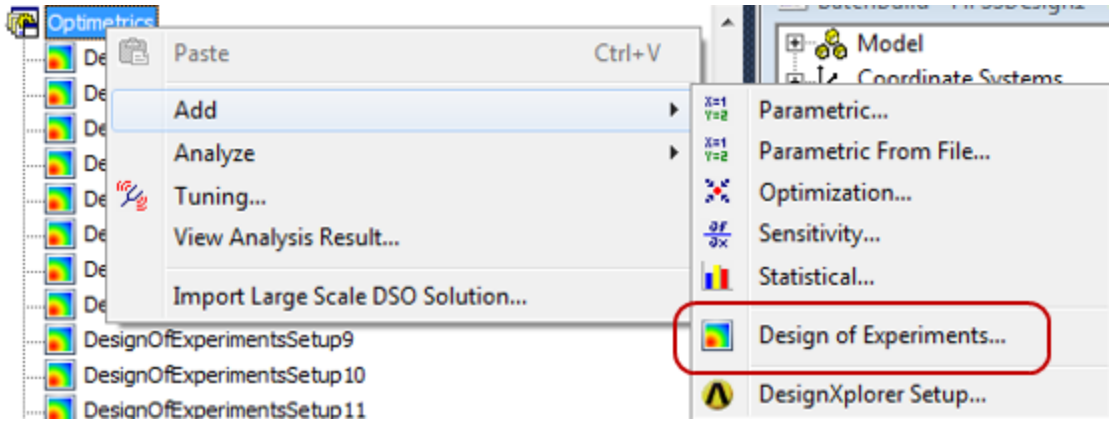
Design of Experiments describes the relationship between the design variables and the performance of the product by using Design of Experiments (DOE), combined with response surfaces. DOE and response surfaces provide all of the information required to achieve Simulation Driven Product Development. Once the variation of the performance with respect to the design variables is known, it becomes easy to understand and identify all changes required to meet the requirements for the product.

The goal is to create a response surface by interpolating through calculated points (a best curve fit). For each design, you can create a response surface for each output parameter. Once the response surfaces are created, you can share the information can in easily understandable terms: curves, surfaces, sensitivities, etc. They can be used at any time during the development of the

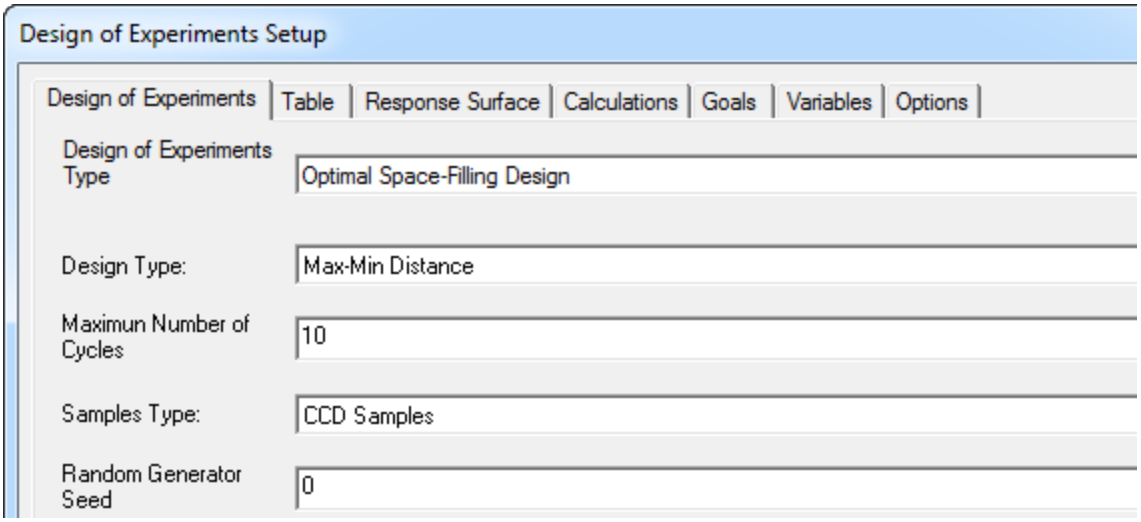
product without requiring additional simulations to test a new configuration.



The Design of Experiments feature is integrated inside Electronics Desktop. Combined with Electronics Desktop‘s distributed solve feature, you can build the response surfaces from the DOE variation table much faster.



Selecting a Design of Experiments under Optimization opens a dialog with several tabs:



In the Design of Experiments setup, you select the DOE type, select the Response Surface, specify goals, view and include variables.

There are a wide range of DOE algorithms or methods available in engineering literature. These techniques all have one common characteristic: they try to locate the sampling points such that the space of random input parameters is explored in the most efficient way, or obtain the required information with a minimum of sampling points. Sample points in efficient locations only reduce the required number of sampling points and increases the accuracy of the response surface generated. For more information on the available types of DOE, see Design of Experiments Types.

Once you have set up your input parameters, you can update the DOE, which submits the generated design points to the analysis system for solution. Design points are solved simultaneously if the analysis system is set up to do so; sequentially, if not. After the solution is complete, you can update the Response Surface cell, which generates response surfaces for each output parameter based on the data in the generated design points.

Note	Requirements and recommendations regarding the number of input parameters vary according to DOE type. For more information, see Number of Input Parameters for DOE Types.

If you change the Design of Experiments type after doing an initial analysis and preview the Design of Experiments Table, any design points generated for the new algorithm that are the same as design points solved for a previous algorithm will appear as up-to-date. Only the design points that are different from any previously submitted design points need to be solved.

You should set up your DOE Properties before generating your DOE Design Point matrix. The following topics describe setting up and solving your Design of Experiments, and viewing the results.

Related Topics

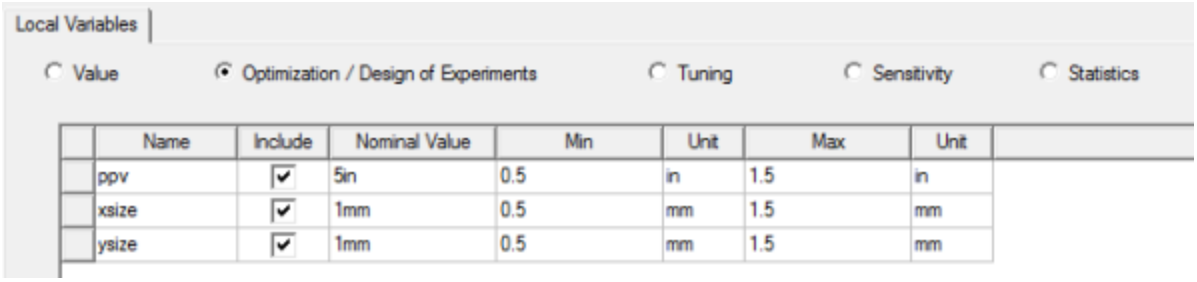
Setting Up Design of Experiments
Viewing the Result for Design of Experiments
Setup Calculations for Optimetrics

Setting Up Design of Experiments

The process for setting up a Design of Experiments analysis is as follows:

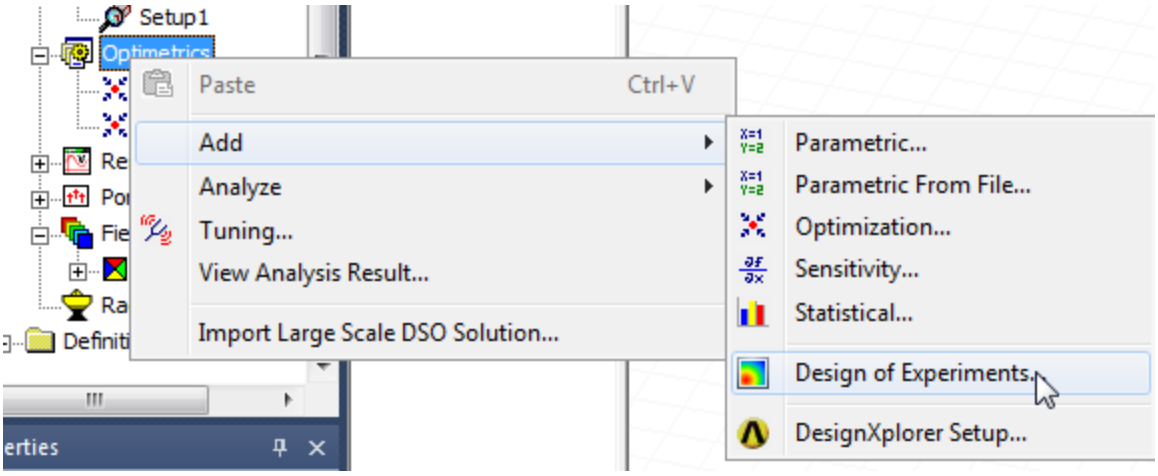
Choose the Variables for Design Exploration

You must define local or project variables as Optimization /Design of Experiments variables for the Design of Experiments setup to include the variable.

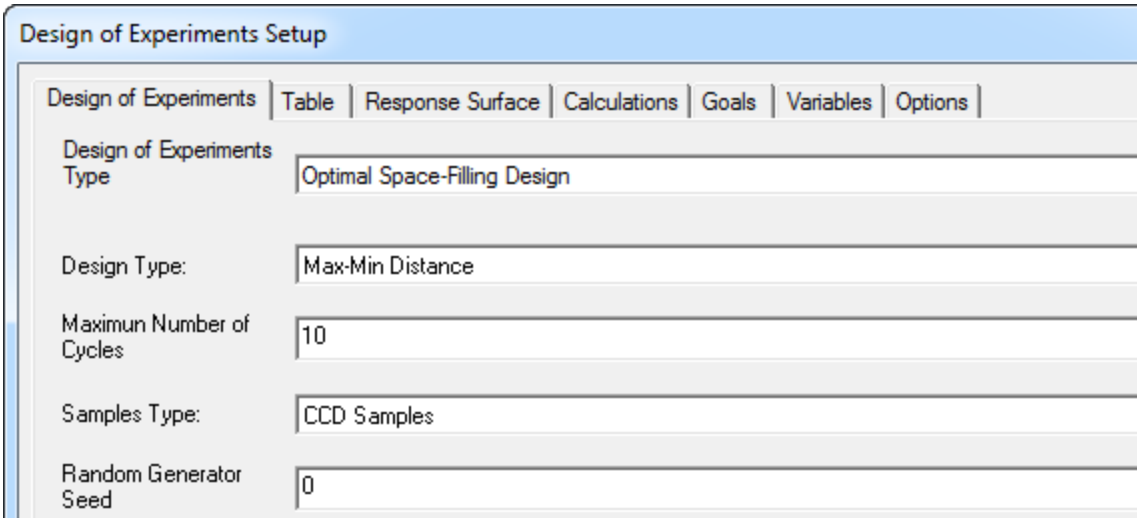


Add the Design of Experiments Setup

Right-click on the Optimetrics icon in the Project tree and select **Add>Design of Experiments**. You can also use **Maxwell 2D** (or **3D**)**>Optimetrics Analysis>Add Design of Experiments**.



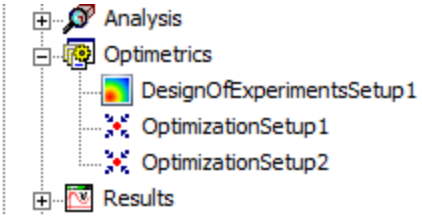
The Design of Experiments Setup dialog appears.



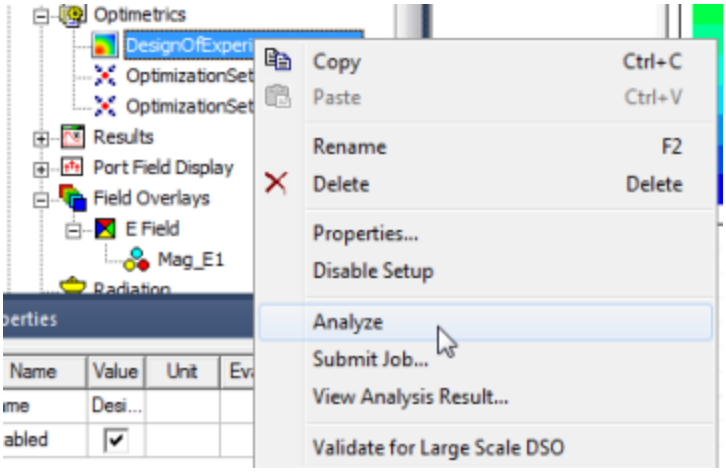
It has tabs for:

- Design of Experiments, where you specify the sampling points and associated parameters.
- Table, which shows the sampling points defined by the Design of Experiments settings you provide. If you select Custom and the Design of Experiments type, you can import data files as well as add or edit rows. You can also Export files of DOE sampling points you have defined by any method.
- Response Surface, which specifies the Response surface type, and Refinement parameters.
- Calculations, which accesses Optimetrics calculations.
- Goals, including cost functions calculations and norm type.
- Variables, previously defined for Optimization/Design of Experiments as Project or Design variables, and whether to include them, treat as discrete, whether to Use Manufacturable Variables, and the available Levels.
- Options, to Save Fields and Mesh, and whether to copy geometrically equivalent meshes.

Once you have set parameters and click **OK**, the **Design of Experiments** setup appears under the Optimetrics icon in the Project tree.



From here you can right click on the DesignOfExperiments setup for the shortcut menu to run **Analyze**, **Submit Job...** or **Validate for Large Scale DSO**.

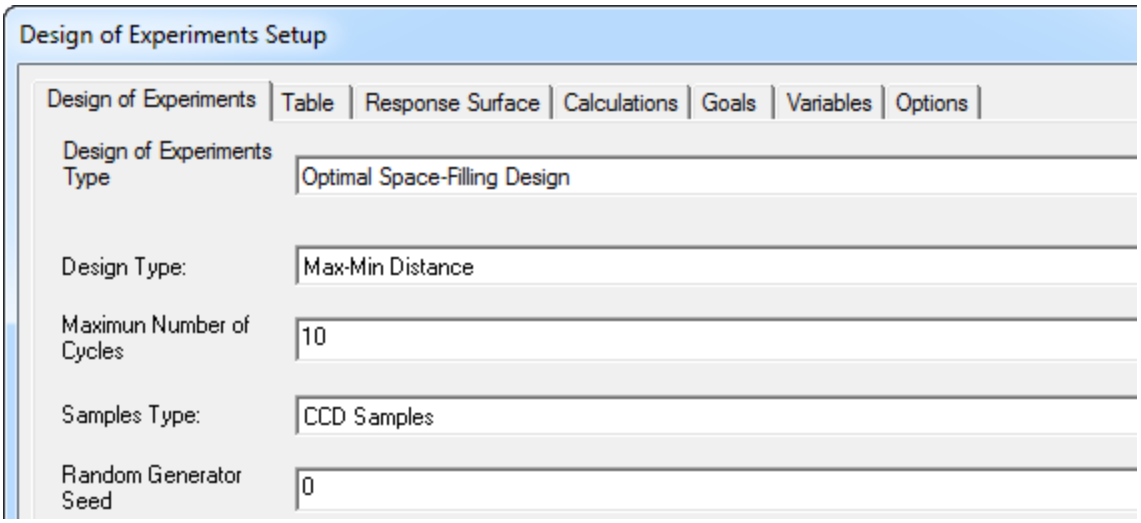


Related Topics

Using Design of Experiments
Design of Experiments Tab
Table Tab for Design of Experiments
Response Surface Tab for Design of Experiments
Variables Tab for Design of Experiments
Setup Calculations for Optimetrics
View Analysis Results for Design of Experiments

Design of Experiments Tab

The Design of Experiments tab in the DOE Setup includes selections for defining the sampling points that define your experiment. Each selection for Design of Experiments type has a different set of associated parameters so the appearance of the dialog changes to show the parameters for your selection.



The Design of Experiments Types available in the Desktop include the following:

Design of Experiments Types	Brief Description (click links for more details)
Optimal Space Filling(Default)	An optimized Latin Hypercube Sampling maximizing distance between experiments. Several design type criteria are available: • Max-Min Distance • Centered L2 • Maximum Entropy Several sampling types available which determine the number of samples in the design: • CCD samples (Central Composite Designs are five level factorial designs that are suitable for calibrating the quadratic response model) • Linear model samples • Pure quadratic model samples • Full quadratic model samples • User-Defined samples You also specify a Maximum Number of Cycles and a Random Generator Seed.
Central Composite Design	Several design types available: • Face-Centered • Rotatable • VIF-Optimality • G-Optimality

Design of Experiments Types	Brief Description (click links for more details)
	• Auto-defined You can also choose a Standard or Enhanced Template.
Box-Behnken	Avoids critical configurations in the corner of the design space. Maximum number of input parameters is 12.
Custom	Lets you customize a DOE matrix, by editing values, adding or removing samples, and/or importing samples from a CSV file. Selecting Custom enables an Import button and the **Table** tab, as well as buttons to **Add editable rows** or **Delete selected rows**. If you previously solved the DOE using one of the other algorithms, those design points are retained and you can add new design points to the table. You can also import and export design points into the custom DOE Table from the Parameter Set.
Latin Hypercube Sampling	Statistical design where no two experiments share input parameters of the same value. The samples Type can be: • CCD Samples • Linear Model Samples • Pure Quadratic Model Samples • Full Quadratic Model Samples • User-Defined Samples, for which you also specify the Number of Samples. For each samples type, you also specify a Random Generator Seed.

The Table tab provides a preview view of the design points defined by your selections.

Related Topics

Setting Up Design Of Experiments

Optimal Space Filling Design (OSF)

The goal in Design of Experiments is to determine the smallest sufficient set of points required to calculate a response surface. Therefore, you choose the type depending on the parametric problem and targeted response surface. The number of points depends on the number of input parameters, or is user defined.

Optimal Space-Filling Design (OSF) creates optimal space filling Design of Experiments (DOE) plans according to some specified criteria. Essentially, OSF is a Latin Hypercube Sampling Design (LHS) that is extended with post-processing. It is initialized as an LHS and then optimized several times, remaining a valid LHS (without points sharing rows or columns) while achieving a more uniform space distribution of points (maximizing the distance between points).

To offset the noise associated with physical experimentation, classical DOE types such as CCD focus on parameter settings near the perimeter of the design region. Because computer simulation is not quite as subject to noise, though, the Optimal Space-Filling (OSF) design is able to distribute the design parameters equally throughout the design space with the objective of gaining the maximum insight into the design with the fewest number of points. This advantage makes it appropriate when a more complex meta-modeling technique such as Kriging, Non-Parametric Regression, or Neural Networks is used.

OSF shares some of the same disadvantages as LHS, though to a lesser degree. Possible disadvantages of an OSF design are:

- When the CCD Samples sample type is selected, a maximum of 20 input parameters is supported.
- Extremes, such as the corners of the design space, are not necessarily covered.
- The selection of too few design points can result in a lower quality of response prediction.

The following properties are available for the OSF DOE type.

- **Design Type**: The following choices are available:
  - **Max-Min Distance** (default): Maximizes the minimum distance between any two points. This strategy ensures that no two points are too close to each other. For a small size of sampling (N), the Max-Min Distance design generally lies on the exterior of the design space and fill in the interior as N becomes larger. Generally, this is the faster algorithm.
  - **Centered L2**: Minimizes the centered L2-discrepancy measure. The discrepancy measure corresponds to the difference between the empirical distribution of the sampling points and the uniform distribution. This means that the centered L2 yields a uniform sampling. This design type is computationally faster than the **Maximum Entropy** type.
  - **Maximum Entropy**: Maximizes the determinant of the covariance matrix of the sampling points to minimize uncertainty in unobserved locations. This option often provides better results for highly correlated design spaces. However, its cost increases non-linearly with the number of input parameters and the number of samples to be generated. Thus, it is recommended only for small parametric problems.
- **Maximum Number of Cycles**: Determines the number of optimization loops the algorithm needs, which in turns determines the discrepancy of the DOE. The optimization is essentially combinatorial, so a large number of cycles slows down the process. However, this makes the discrepancy of the DOE smaller. For practical purposes, 10 cycles is generally good for up to 20 variables. The value must be greater than 0. The default is 10.
  - **Samples Type**: Determines the number of DOE points the algorithm should generate. This option is suggested if you have some advanced knowledge about the nature of the metamodel. The following choices are available:
    - **CCD Samples** (default): Supports a maximum of 20 inputs. Generates the same number of samples a CCD DOE would generate for the same number of inputs. You

can use this to generate a space filling design that has the same cost as a corresponding CCD design.

- **Linear Model Samples**: Generates the number of samples as needed for a linear metamodel.
- **Pure Quadratic Model Samples**: Generates the number of samples as needed for a pure quadratic metamodel (no cross terms).
- **Full Quadratic Samples**: Generates the number of samples needed to generate a full quadratic model.
- **User-Defined Samples**: Specify the desired number of samples.

- **Seed Value**: Set the value used to initialize the random number generator invoked internally by the LHS algorithm. Although the generation of a starting point is random, the seed value consistently results in a specific LHS. This property allows you to generate different samplings by changing the value or regenerate the same sampling by keeping the same value. The default is 0.
- **Number of Samples**: Enabled when **Samples Type** is set to **User-Defined Samples**. Specifies the default number of samples. The default is 10.

Related Topics

Setting Up Design Of Experiments

Central Composite Design (CCD)

The goal in Design of Experiments is to determine the smallest sufficient set of points required to calculate a response surface. Therefore, you choose the type depending on the parametric problem and targeted response surface. The number of points depends on the number of input parameters, or is user defined.

Central Composite Design (CCD) provides a screening set to determine the overall trends of the metamodel to better guide the choice of options in Optimal Space-Filling Design. The CCD DOE type supports a maximum of 20 input parameters.

The following properties are available for the CCD DOE type:

- **Design Type**: By specifying the **Design Type** for CCD, you can help to improve the response surface fit for DOE studies. For each CCD type, the alpha value is defined as the location of the sampling point that accounts for all quadratic main effects. The following CCD design types are available:
  - **Face-Centered**: A three-level design with no rotatability. The alpha value equals 1.0. A Template Type setting automatically appears, with Standard and Enhanced options. Choose Enhanced for a possible better fit for the response surfaces.
  - **Rotatable**: A five-level design that includes rotatability. The alpha value is calculated based on the number of input variables and a fraction of the factorial part. A design with rotatability has the same variance of the fitted value regardless of the direction from the center point.

- **VIF-Optimality**: A five-level design in which the alpha value is calculated by minimizing a measure of non-orthogonality known as the Variance Inflation Factor (VIF). The more highly correlated the input variable with one or more terms in a regression model, the higher the VIF.
- **G-Optimality**: Minimizes a measure of the expected error in a prediction and minimizes the largest expected variance of prediction over the region of interest.
- **Auto-Defined**: Design exploration automatically selects the Design Type based on the number of input variables. Use of this option is recommended for most cases as it automatically switches between the G-Optimality if the number of input variables is 5 or VIF-Optimality otherwise.

  However, you can use the Rotatable design if the default option does not provide good values for the Goodness of Fit from the response surface plots. Additionally, you can use the Enhanced template if the default Standard template does not fit the response surfaces well.

- **Template Type**: Enabled for the Rotatable and Face-Centered design types. The following options are available:
  - **Standard**
  - **Enhanced**: Choose this option for a possible better fit for the response surfaces

Related Topics

Setting Up Design Of Experiments

Box Behnken Design (CCD)

The goal in Design of Experiments is to determine the smallest sufficient set of points required to calculate a response surface. Therefore, you choose the type depending on the parametric problem and targeted response surface. The number of points depends on the number of input parameters, or is user defined.

A **Box-Behnken Design** is a three-level quadratic design that does not contain fractional factorial design. The sample combinations are treated in such a way that they are located at midpoints of edges formed by any two factors. The design is rotatable (or in cases, nearly rotatable).

One advantage of a Box-Behnken design is that it requires fewer design points than a full factorial CCD and generally requires fewer design points than a fractional factorial CCD. Additionally, a Box-Behnken Design avoids extremes, allowing you to work around extreme factor combinations. Consider using the Box-Behnken Design DOE type if your project has parametric extremes (for example, has extreme parameter values in corners that are difficult to build). Since the Box-Behnken DOE doesn't have corners and does not combine parametric extremes, it can reduce the risk of update failures.

Possible disadvantages of a Box-Behnken design are:

- Prediction at the corners of the design space is poor and that there are only three levels per parameter.

- A maximum of 12 input parameters is supported.

No additional properties are available for the Box-Behnken Design DOE type.

Related Topics

Setting Up Design Of Experiments

Custom DOE Type

The goal in Design of Experiments is to determine the smallest sufficient set of points required to calculate a response surface. Therefore, you choose the type depending on the parametric problem and targeted response surface. The number of points depends on the number of input parameters, or is user defined.

The **Custom DOE** type allows for definition of a custom DOE Table. You can manually add new design points, entering the input and (optionally) output parameter values directly into the table. If you previously solved the DOE using one of the other algorithms, those design points are retained and you can add new design points to the table. You can also import and export design points into the custom DOE Table from the Parameter Set.

You can change the edition mode of the DOE table to edit the output parameter values. You can also copy and paste data and import data from a CSV file by right-clicking and selecting Import Design Points.

Related Topics

Setting Up Design Of Experiments

Latin Hypercube Sampling

The goal in Design of Experiments is to determine the smallest sufficient set of points required to calculate a response surface. Therefore, you choose the type depending on the parametric problem and targeted response surface. The number of points depends on the number of input parameters, or is user defined.

In the **Latin Hypercube Sampling** Design DOE type, the DOE is generated by the LHS algorithm, an advanced form of the Monte Carlo sampling method that avoids clustering samples. In a Latin Hypercube Sampling, the points are randomly generated in a square grid across the design space, but no two points share the same value. This means that no point shares a row or a column of the grid with any other point.

Possible disadvantages of an LHS design are:

- When the CCD Samples sample type is selected, a maximum of 20 input parameters is supported. For more information, see Number of Input Parameters for DOE Types.
- Extremes, such as the corners of the design space, are not necessarily covered. Additionally, the selection of too few design points can result in a lower quality of response prediction.

Note: The Optimal Space-Filling Design DOE type is an LHS design that is extended with post-processing.

The following properties are available for the LHS DOE type:

- **Samples Type**: Determines the number of DOE points the algorithm should generate. This option is suggested if you have some advanced knowledge about the nature of the metamodel. The following choices are available:
  - **CCD Samples** (default): Supports a maximum of 20 inputs. Generates the same number of samples a CCD DOE would generate for the same number of inputs. You can use this to generate a space filling design that has the same cost as a corresponding CCD design.
  - **Linear Model Samples**: Generates the number of samples as needed for a linear metamodel.
  - **Pure Quadratic Model Samples**: Generates the number of samples as needed for a pure quadratic metamodel (no cross terms).
  - **Full Quadratic Samples**: Generates the number of samples needed to generate a full quadratic model.
  - **User-Defined Samples**: Specify the desired number of samples.
- **Seed Value**: Set the value used to initialize the random number generator invoked internally by the LHS algorithm. Although the generation of a starting point is random, the seed value consistently results in a specific LHS. This property allows you to generate different LHS samplings (by changing the value) or to regenerate the same LHS sampling (by keeping the same value). The default is 0.
- **Number of Samples**: Enabled when **Samples Type** is set to **User-Defined Samples**. Specifies the default number of samples. The default is 10.

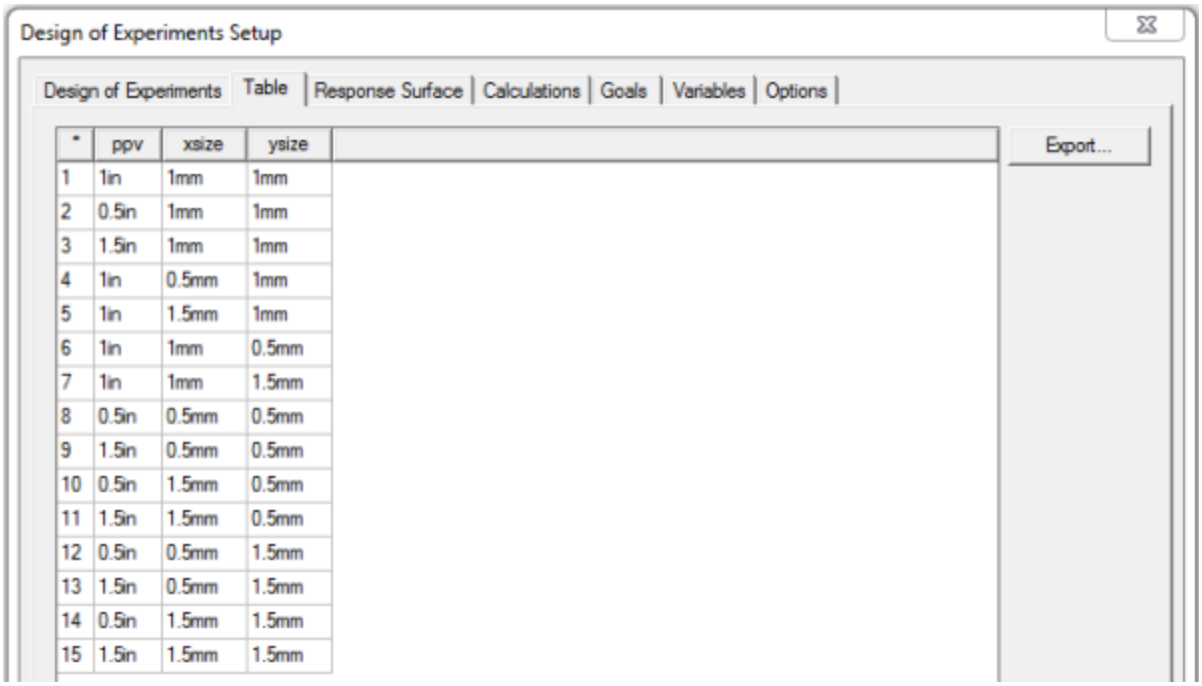
Related Topics

Setting Up Design Of Experiments

Table Tab for Design of Experiments

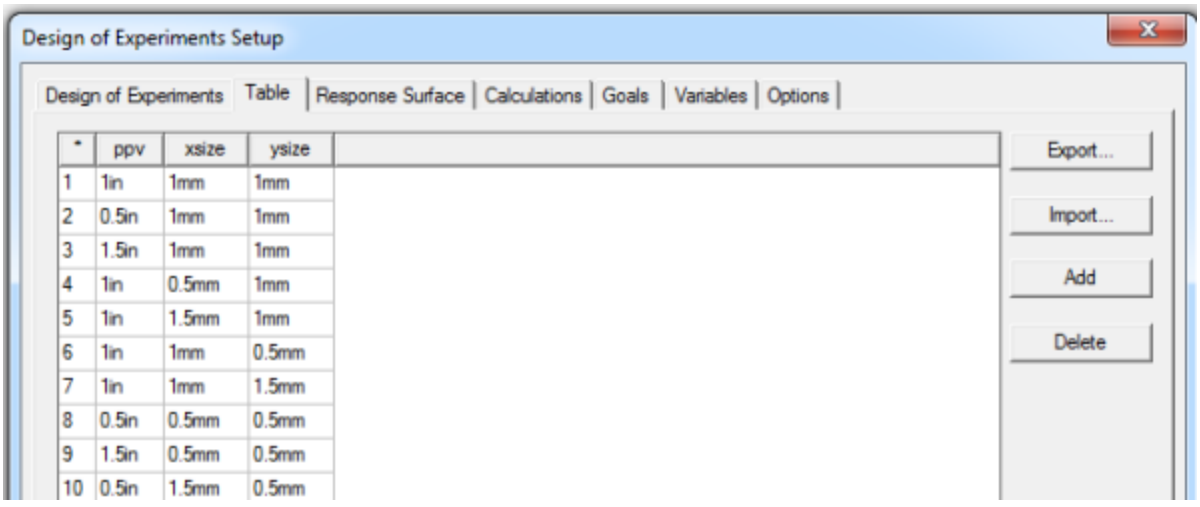
The Table tab for the **Design of Experiments Setup** dialog displays a preview of the design points designed by your selections on the **Design of Experiments** tab. There is one column for

each defined variable. The **Export** button lets you create a file of the table in a format you specify.



*	ppv	xsize	ysize
1	1in	1mm	1mm
2	0.5in	1mm	1mm
3	1.5in	1mm	1mm
4	1in	0.5mm	1mm
5	1in	1.5mm	1mm
6	1in	1mm	0.5mm
7	1in	1mm	1.5mm
8	0.5in	0.5mm	0.5mm
9	1.5in	0.5mm	0.5mm
10	0.5in	1.5mm	0.5mm
11	1.5in	1.5mm	0.5mm
12	0.5in	0.5mm	1.5mm
13	1.5in	0.5mm	1.5mm
14	0.5in	1.5mm	1.5mm
15	1.5in	1.5mm	1.5mm

If you have specified Custom as the Design of Experiments Type, the table is editable and the Table tab includes buttons for Add and Delete Rows. The added Rows are editable. You can add new rows by entering values in the * row of the table. You enter values in the input parameter columns. Once you have entered a value in one column in the * row, the row is added to the table and the values for the remaining input parameters are set to the initial values of the parameters. You can then edit that row in the table and change any of the other input parameter values if needed. Output parameter values are then calculated when the design is solved updated.



*	ppv	xsize	ysize
1	1in	1mm	1mm
2	0.5in	1mm	1mm
3	1.5in	1mm	1mm
4	1in	0.5mm	1mm
5	1in	1.5mm	1mm
6	1in	1mm	0.5mm
7	1in	1mm	1.5mm
8	0.5in	0.5mm	0.5mm
9	1.5in	0.5mm	0.5mm
10	0.5in	1.5mm	0.5mm

Depending on the context, the tables are read-only and filled automatically or they are partially or completely editable. The background color of a cell indicates if it is editable or not:

- A gray background indicates a read-only cell
- A white background indicates an editable cell

Output parameter values calculated from a simulation (a design point update) are displayed in black text.

The Custom Table view also includes an Import button. Import and Export files can be:

- Comma delimited data files, (*.csv)
- Tab delimited data files, (*.tab)
- Ansoft Plot Data files, (.dat)
- Post Processor format data files, (*.txt)

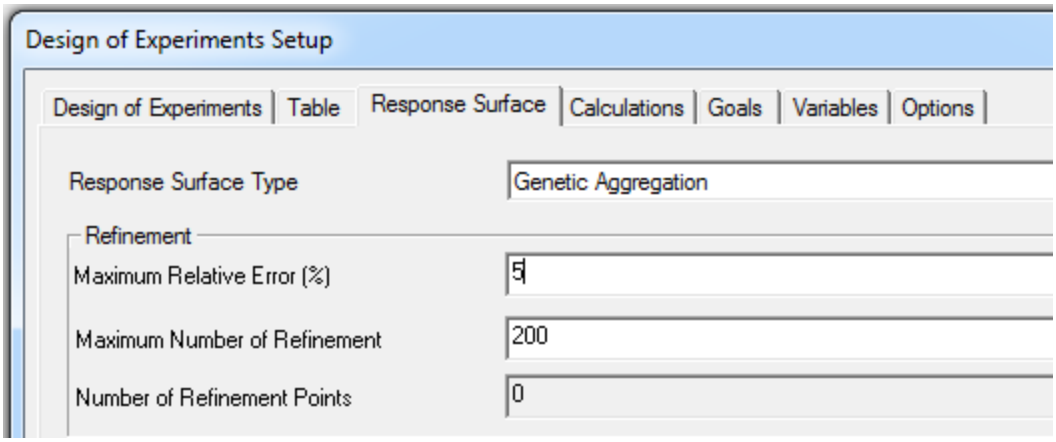
The Table updates automatically when you change your selections on the **Design of Experiments** tab.

Related Topics

Custom DOE Type

Response Surface Tab for Design of Experiments

The Response Surface tab for the **Design of Experiments Setup** dialog lets you select the Response Surface type as Genetic Aggregation or Standard Response Surface.



The selection here specifies the refinement applied to the initial Design of Experiments. The Genetic Aggregation Response Surface finds the best possible response surface for each output automatically by combining

- Metamodels
- Settings

- Kernel Variation
- Polynomial Regression

For each output, a Fitness factor works to minimize error, including cross-validation errors. The automatic refinement adds design points to the DOE until the response surface accuracy meets user requirements. You can specify requirements for:

- **Maximum Relative Error %** – Applies to all output calculations. Empty this field to turn off Auto refinement. (Note: Workbench does not have this field.)
- **Maximum Number of Refinement** attempts
- **Number of Refinement Points** – Read-only property indicating the number of existing refinement points.

If you check **Show Advanced Options** at the bottom of the Setup dialog box, you can also specify the **Crowding Distance Separation Percentage** which determines the minimum allowable distance between new refinement points, implemented as a constraint in the search for refinement points.

You also have the option of selecting **Standard Response Surface- Full 2nd Order Polynomial** as the Response Surface Type.

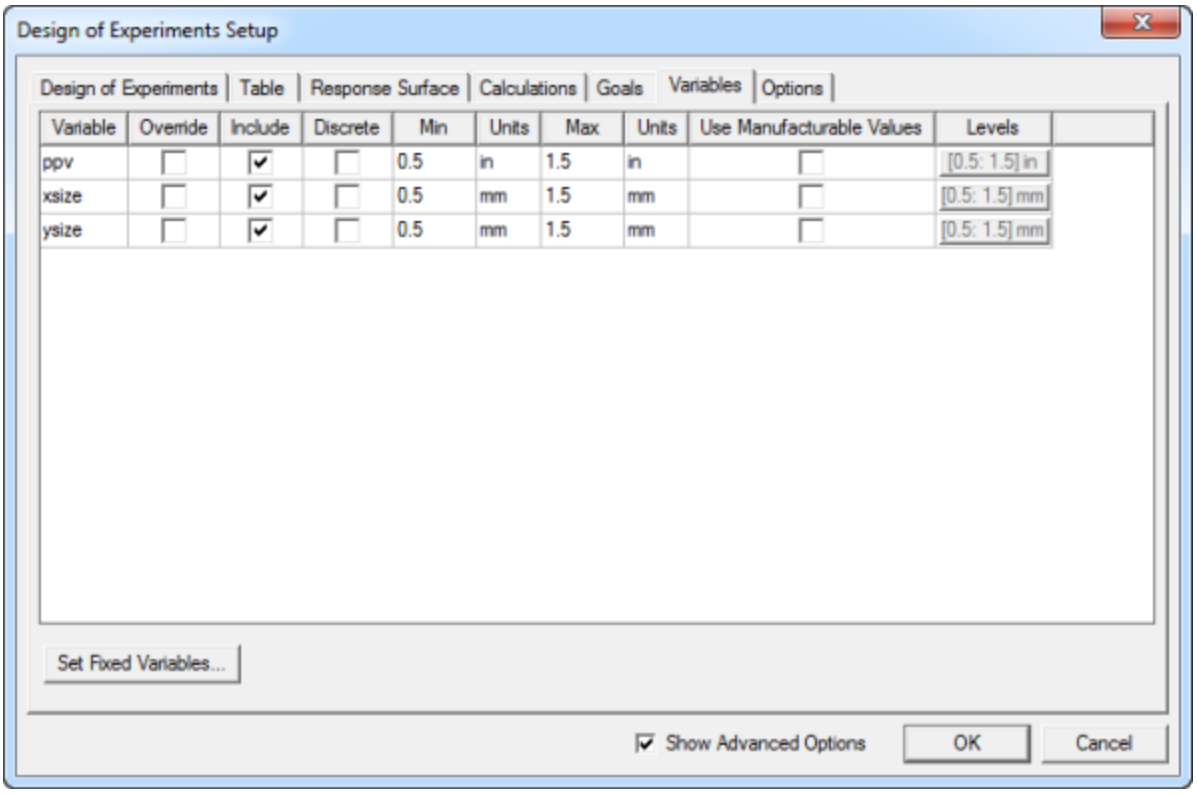
After you have completed an analysis you can view the generated plot.

Related Topics

Setting Up Design Of Experiments

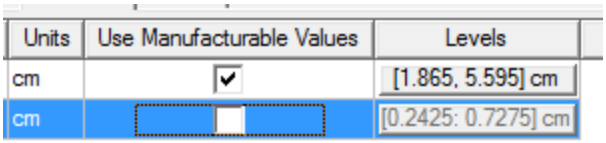
Variables Tab for Design of Experiments

The Variables tab for the **Design of Experiments Setup** dialog displays list of the variables defined for the Design and Project as Optimetrics/Design of Experiments variables.



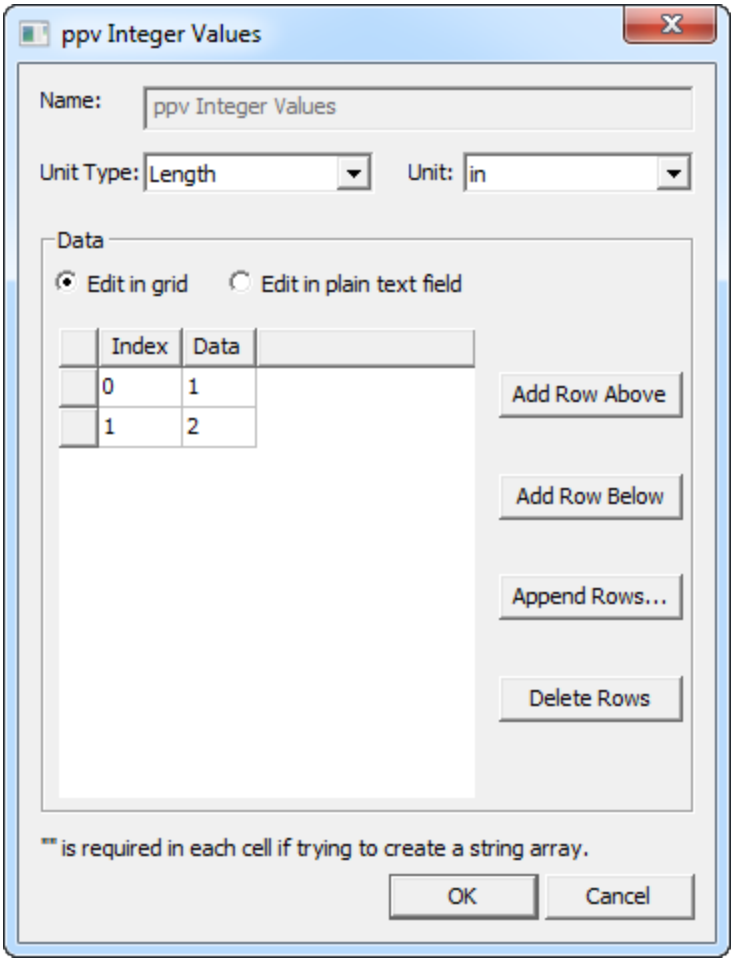
The columns list the Variable names, the current Min and Max Values and Units, and provides options for the following:

- Override--check this to override the current design value. Unchecking this causes a dialog to appear asking you to confirm the return to the design value.
- Include--whether to include the variable in an analysis.
- Discrete--**Discrete Variables** physically represent different configurations or states of the model. When checkbox in the Discrete column is checked the button in the “Levels” column will be enabled.

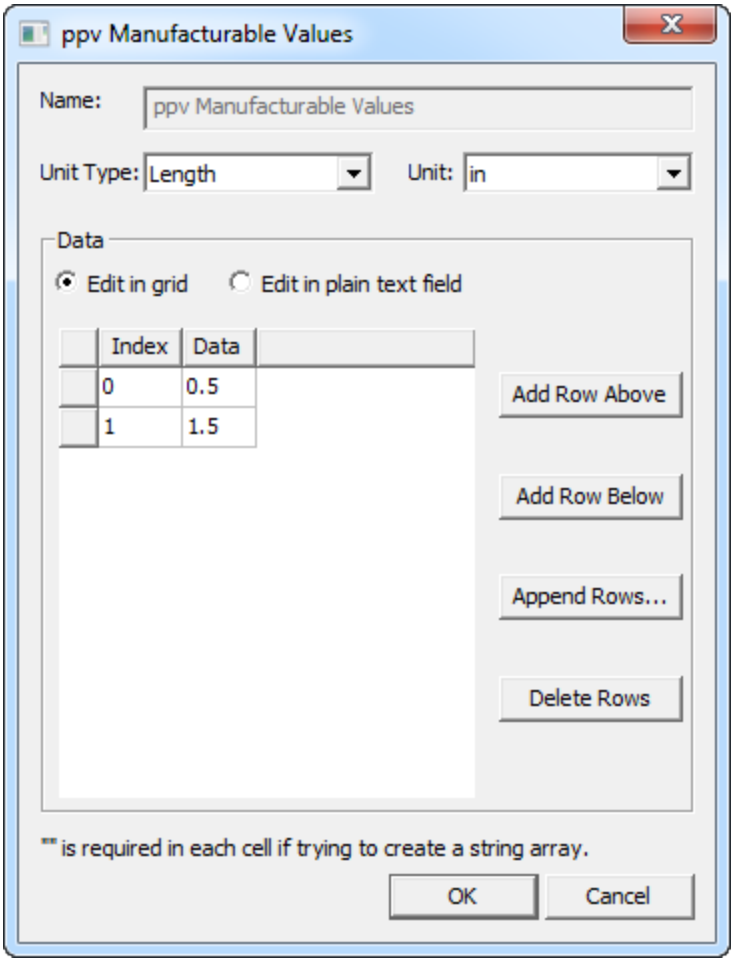


Levels show the variable values. For continuous variable, Levels is determined by the min/max columns. If you check **Use Manufacturable Values**, Levels lists the Manufacturable values (for example, [min, va, v2, v3, max]). For Integer variables Levels can be a subset of the min/max range. The discrete values can be bounded by a min/man range and/or manufacturable values.

- Click the Levels button for the row to edit the Discrete values. An edit dialog for the variable appears.



- **Continuous Variables** physically vary in a continuous manner between a lower and an upper bound (min/max) defined by the user. With continuous Variables, you can also apply a Manufacturable Values filter to the Variable. Manufacturable Values represent real-world manufacturing or production constraints. The min man values will be upper and lower constraints on the available manufacturable values bounded by the range.
- When check box in the “Use Manufacturable Values” column is checked the button in the “Levels” column is enabled. Click the button to edit the values. An edit dialog for the variable appears.



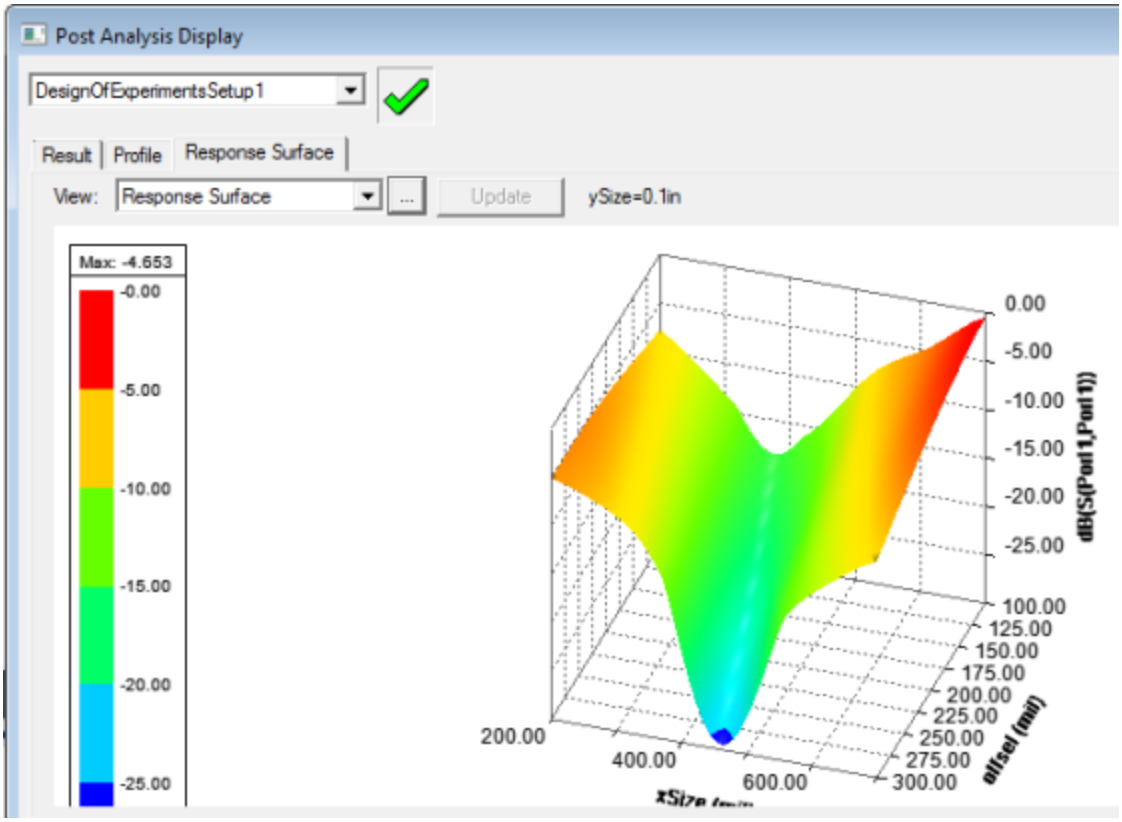
Related Topics

Setting Up Design Of Experiments

View Analysis Result for Design of Experiments

After the Analysis has competed, you can click **View Analysis Result**. This opens a dialog that includes a **Result** tab that lists the variations and variable values, a solution **Profile** tab with start, stop, time elapsed, and machines used, as well as a **Response Surface** tab. When you include more than one variable in the setup, response surface view is available. You can choose any two variables as the X, Y axis, and choose an output calculation as the Z axis, by clicking the "..."

button next to the view list box.



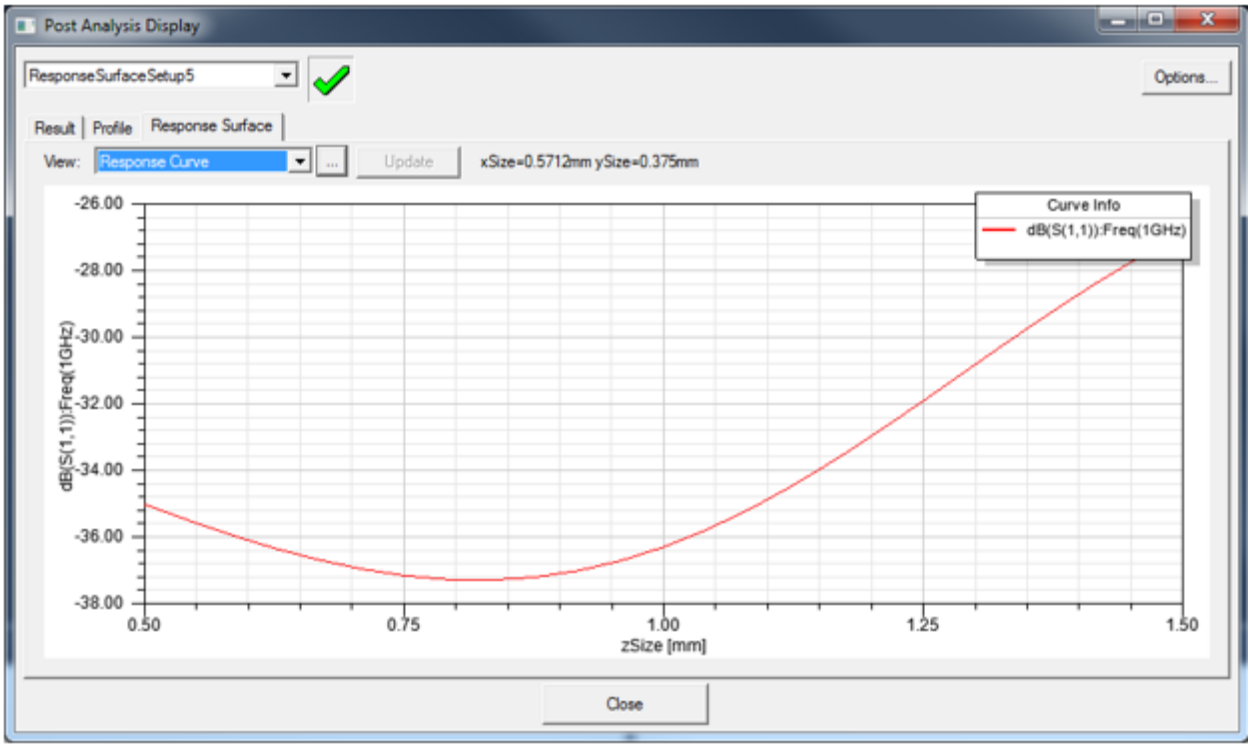
From the Response Surface tab, the “View” list box lets you select all available views of the selected response-surface-setup.

- Min Max Search
- Refinement points table
- Response points table
- Verification points table
- Goodness of Fit
- Response Curve (see below)
- Response Curve (2D Slices, see below)
- Response Surface

The **Update** push button is disabled when the response surface is up to date. After the setup, if you modify a verification point or refinement points, it is enabled. Click this button to re-generate the response-surface with new settings. It may start new simulations if any of the design points in the DOE, refinement points, or verification points has not been solved.

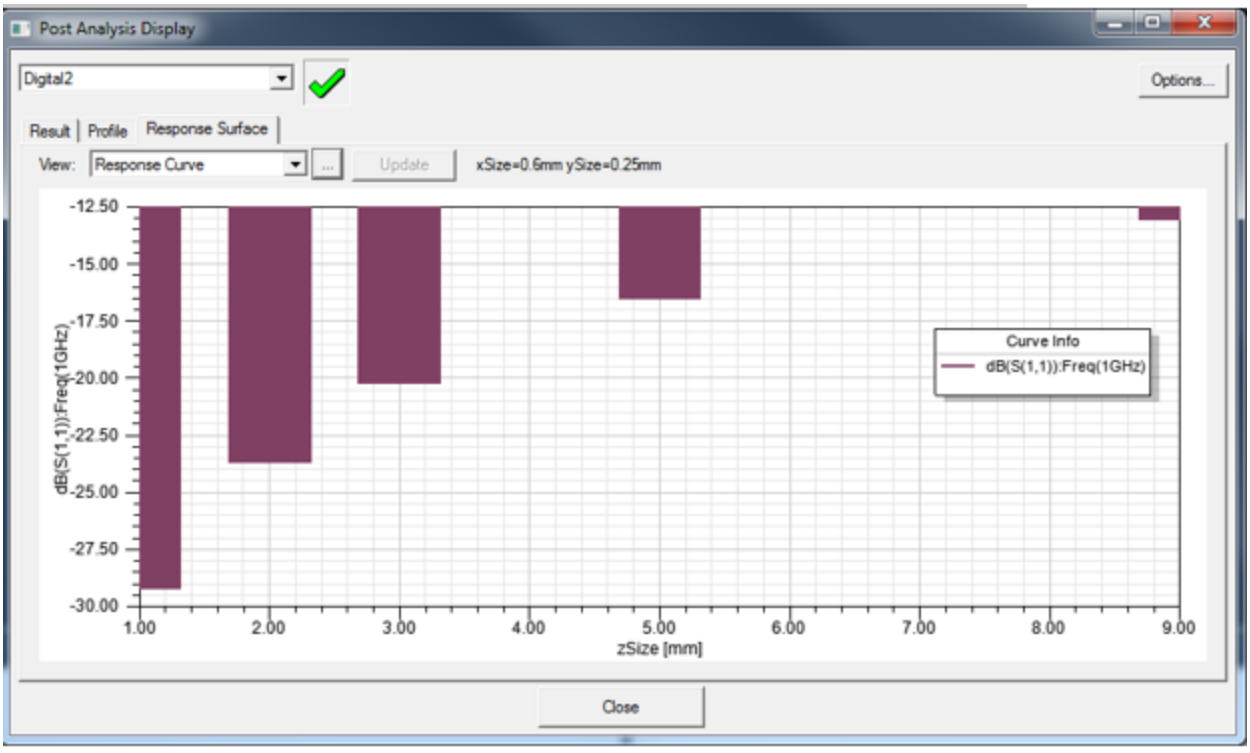
Response Curve

When Continuous Variable is chosen as the X-axis, a Continuous XY plot will be shown.



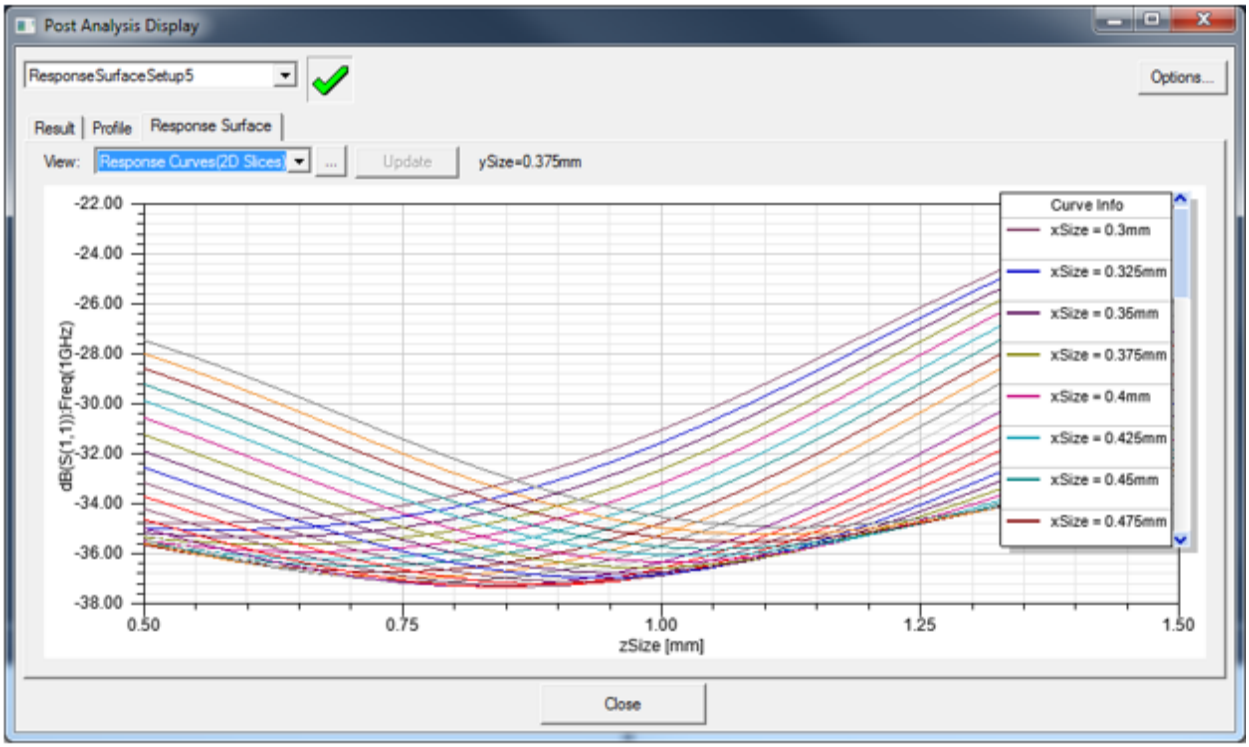
Discrete Variable

If you choose Discrete Variable as the X-axis, a bar chart plot is shown:

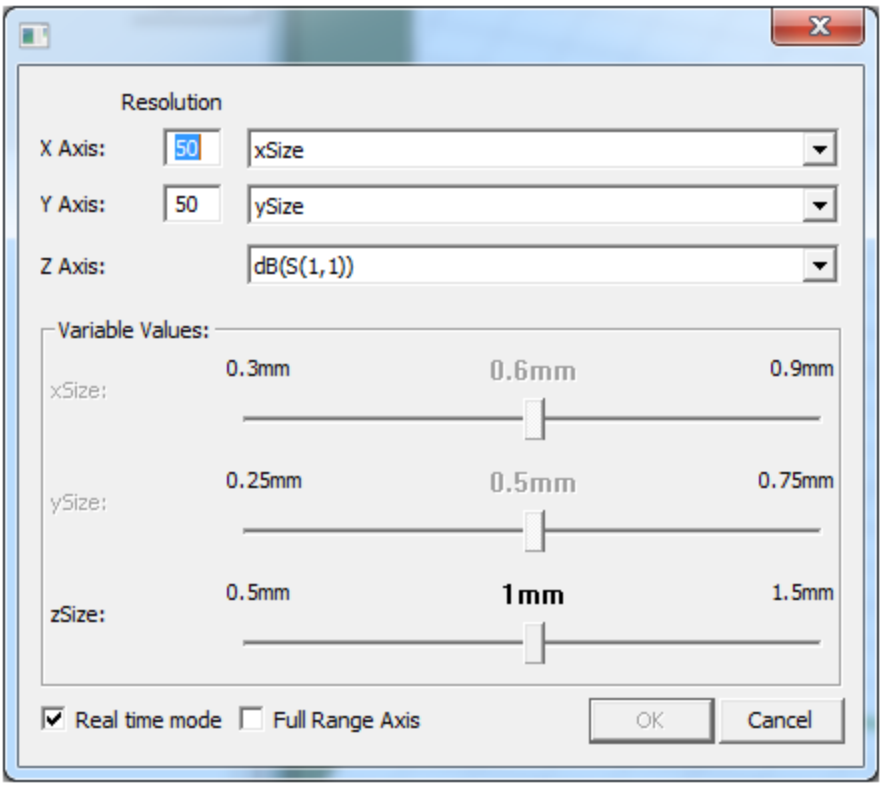


Response Curves (2D Slices)

If you have included more than one variable in the setup, the Response Curves (2D) view is available:



Clicking the ellipsis button [...] by the View for Response Surface opens a dialog that lets you adjust the variables selected and the values applied.



Tuning a Response Curve

For the X Axis and Y Axis, you can specify a Resolution, and the variable to use. For the Z Axis you can select the Cost or calculation. For variables not selected for the X and Y axis, a slider is enabled that lets you adjust the value to see the effect on the response curve plot. You can enable or disable Real time mode by using the check box at the lower left.

Full Range Axis check box

When it is checked all Axes are set to their maximum ranges, and the ranges won't be changed while tuning unless you change the axis variable.

When it is unchecked, the Y(2D)/Z(3D) axis range is auto updated to fit the curve/surface.

Exporting Response Curve Data

You can Export the response curve data as a table in the following formats:

- Comma delimited data files, (*.csv)
- Tab delimited data files, (*.tab)
- Ansoft Plot data files, (*.dat)
- Post Processor format data files, (*.txt)

  Exported files can be imported into a report.

Related Topics

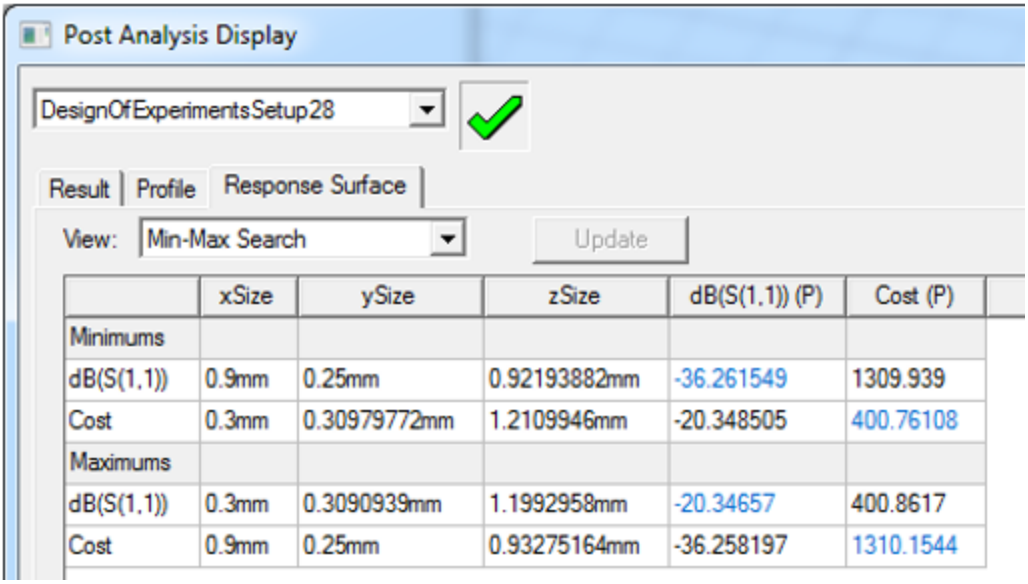
Using Design of Experiments

Setting Up Design of Experiments

Min-Max Search View for Design of Experiments Result

From the Response Surface tab, the “View” list box lets you select all available views of the selected response-surface-setup.

The **Min-Max Search** view examines the entire output parameter space from a response surface to approximate the minimum and maximum values of each output parameter. When you select a Min/Max row the **Apply** button is enabled, and you can then apply the selected variation variable values to the variables’ nominal values.



	xSize	ySize	zSize	dB(S(1,1)) (P)	Cost (P)
Minimums					
dB(S(1,1))	0.9mm	0.25mm	0.92193882mm	-36.261549	1309.939
Cost	0.3mm	0.30979772mm	1.2109946mm	-20.348505	400.76108
Maximums					
dB(S(1,1))	0.3mm	0.3090939mm	1.1992958mm	-20.34657	400.8617
Cost	0.9mm	0.25mm	0.93275164mm	-36.258197	1310.1544

You can Export the Table in the following formats:

- Comma delimited data files, (*.csv)
- Tab delimited data files, (*.tab)
- Ansoft Plot data files, (*.dat)
- Post Processor format data files, (*.txt)

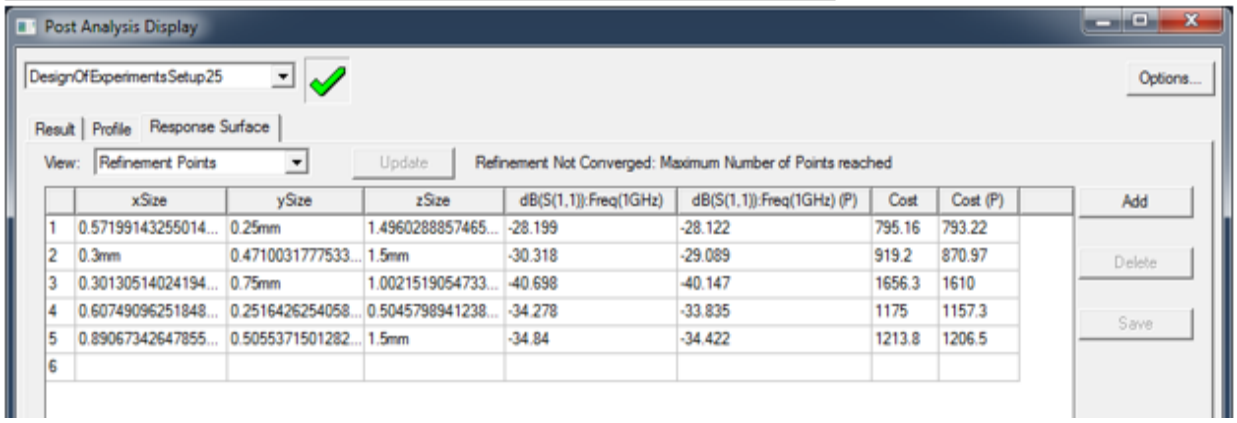
Related Topics

Setting Up Design of Experiments

Using Design of Experiments

Refinement Points Table

From the Response Surface tab, the “View” list box lets you select all available views of the selected response-surface-setup. All refinement points are shown in the Refinement Points table:



Refinement points are points added to your model to enrich and improve your response surface. They can either be generated automatically with the response surface update or added manually, as described in **Performing a Manual Refinement**. As with design points, DesignXplorer must perform a design point update (a "real solve") in order to obtain the output parameters for the refinement points.

Upon update, the refinement points are used to build the response surface and are taken into account for the generation of verification points. Along with DOE points, refinement points are also used as "learning points" for Goodness of Fit calculations.

Performing a Manual Refinement

Manual refinement is a way to force the response surface to take into account points of your choice, in addition to the points already in the Design of Experiments. You can insert a refinement point in the Refinement Points table, and do not need to do an initial solve of the response surface (without the refinement point) before updating your response surface with your manual refinement. Manual refinement is available for all response surface types except for Sparse Grid.

You can add, delete, or modify refinement points by clicking the **Add** or **Delete** buttons, or by modifying point values directly in the grid.

Manual Refinement Point can be inserted from the Response point table and Verification Points Table.

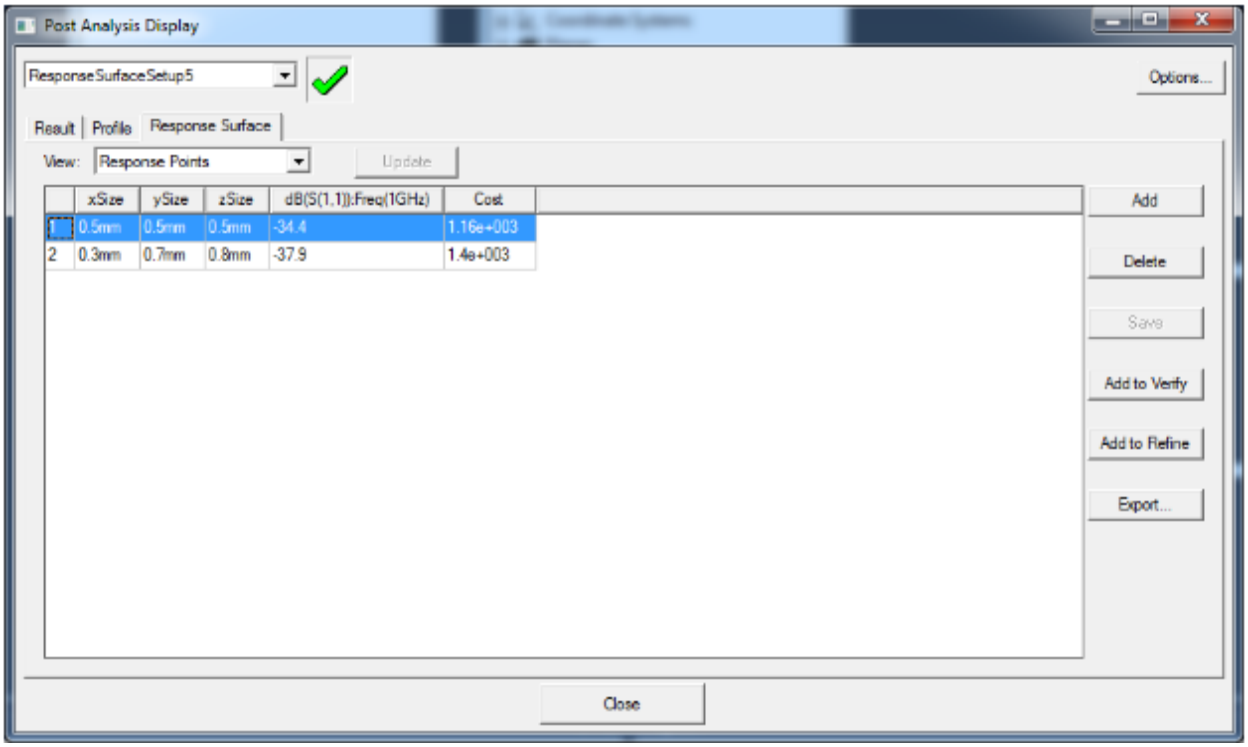
Related Topics

Setting Up Design of Experiments

Using Design of Experiments

Response Points Table

From the Response Surface tab, the “View” list box lets you select all available views of the selected response-surface-setup. All Response points are shown in the Response Points table:



A response point is defined by a snapshot of variable values where output calculation values were calculated in Ansys DesignXplorer from a response surface. As such, the output calculation(or cost) values are approximate and calculated from response surfaces.

You can add, delete, save, or export response points by using the command buttons in the dialog or you can modify response points manually by modifying point values directly in the grid.

Click **Add to Verify** or **Add to Refine** buttons to insert the selected response point to the verification table or refinement table.

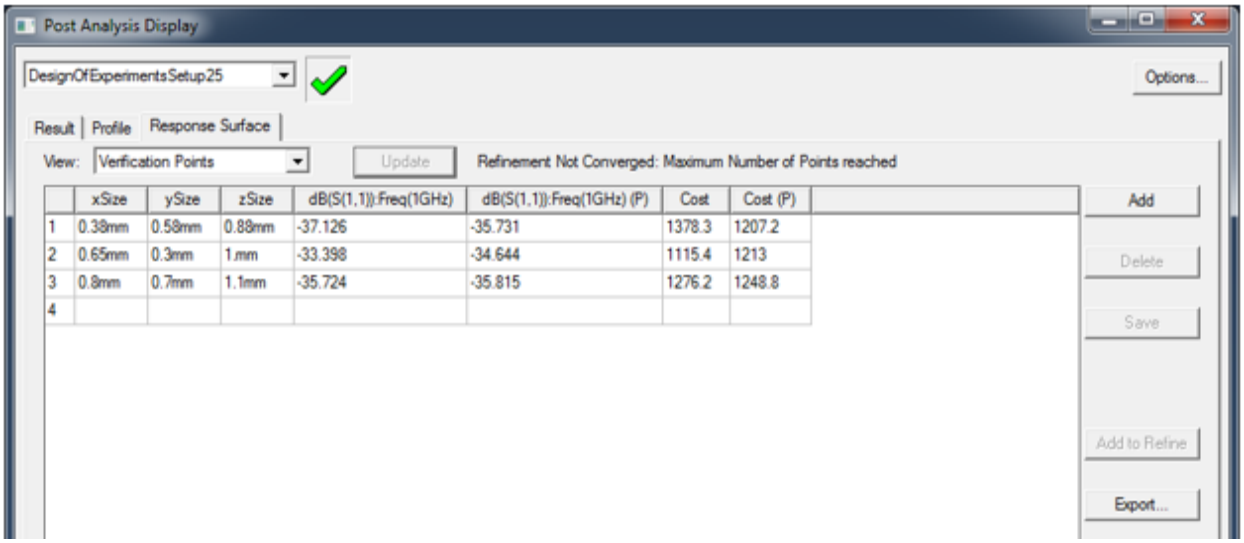
Related Topics

Setting Up Design of Experiments

Using Design of Experiments

Verification Points Table

From the Response Surface tab, the “View” list box lets you select all available views of the selected response-surface-setup. All Verification points are shown in the Verification Points table:



Verification points enable you to verify that the response surface accurately approximates the output parameter values; they compare the predicted and observed values of the output parameters.

You can add, delete and modify Verification points manually.

- Same as add/delete/modify refinement points.
- Insert from Response Points Table.

Click the **Add to Refine** button to insert the selected response point to the refinement table.

A design point update (that is, a "real solve") calculates each verification point. These verification point results are then compared with the response surface predictions and the difference is calculated.

Verification points are useful in validating any type of response surface. In particular, however, you should always use verification points to validate the accuracy of interpolated response surfaces, such as Kriging or Sparse Grid.

You can add, delete, save, or export verification points by using the command buttons in the dialog or you can modify verification points manually by modifying point values directly in the grid.

Related Topics

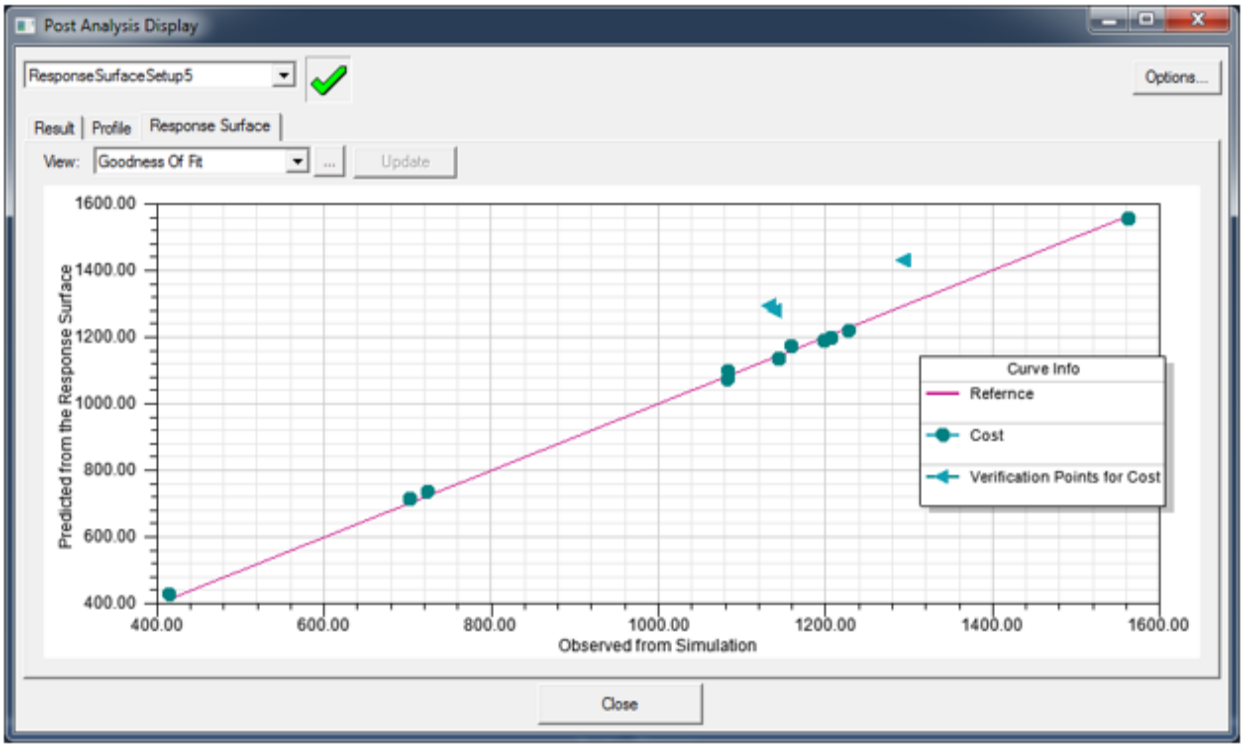
Setting Up Design of Experiments

Using Design of Experiments

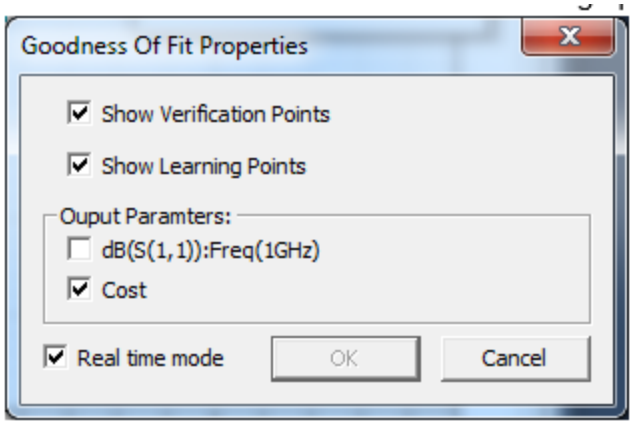
Goodness of Fit (Predicted vs Observed Chart)

From the Response Surface tab, the “View” list box lets you select all available views of the selected response-surface-setup.Response surfaces are built from design points in the Design of

Experiments (DOE) and refinement points (collectively, called "learning points"). The Goodness of Fit calculations compare the response surface outputs with the DOE results used to create them. The closer the points are to the diagonal line, the better the response surface fits the points.



You can view Goodness of Fit information for any of the output parameters in a response surface. To do so click the ellipsis button [...] button to bring up this dialog:



Related Topics

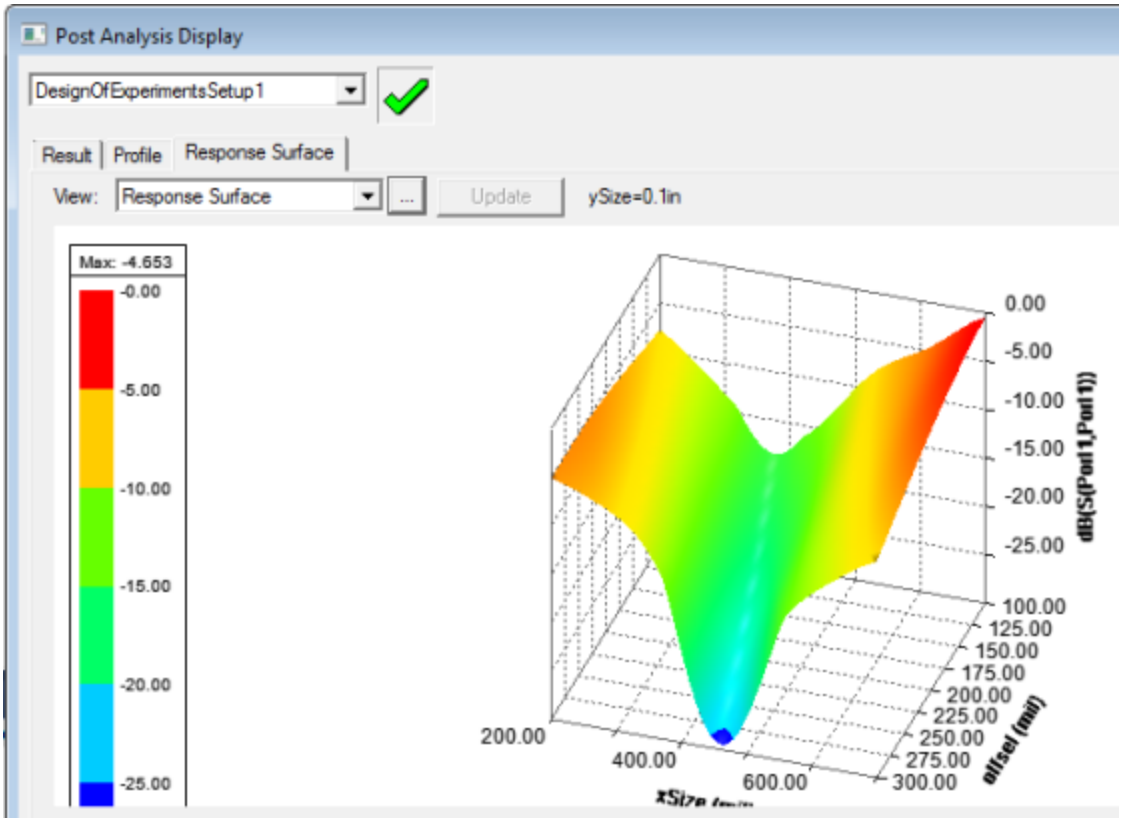
Setting Up Design of Experiments

Using Design of Experiments

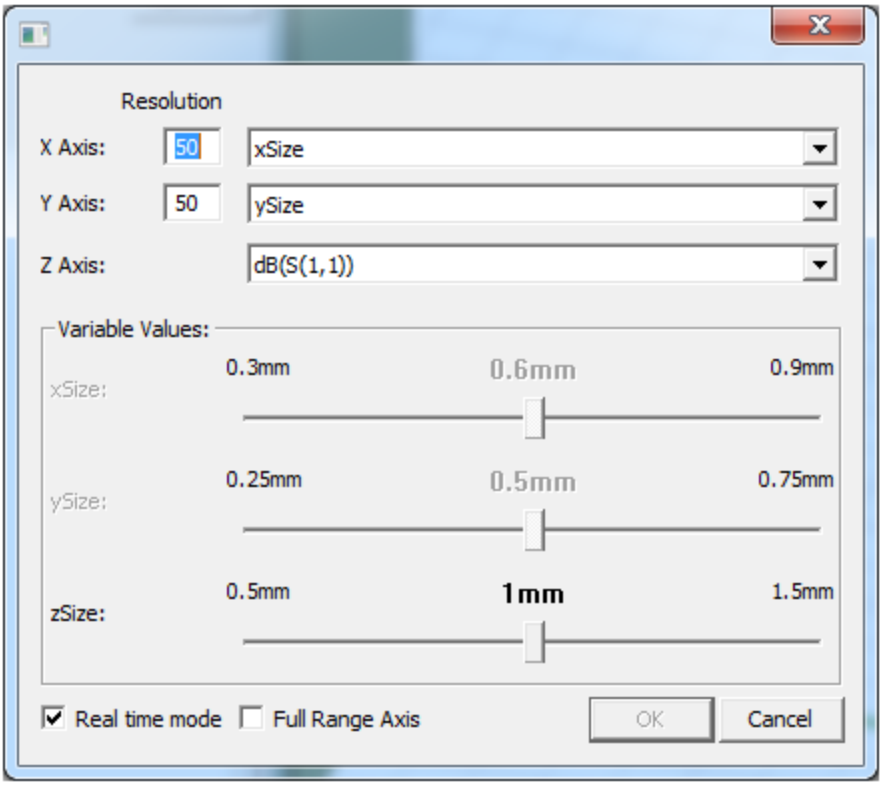
Response Surface Results Design of Experiments Result

When more than one variable is included in the setup, the Response Surface view is available. You can choose any two variables as the X, Y axis, and choose an output calculation as the Z axis, by clicking the "..." button next to the view list box. From the **Response Surface** tab of the Design of Experiments **Post Analysis Display** dialog box, the "View" list box lets you select all available views of the selected response-surface-setup.

Response surfaces are functions of varying natures in which the output parameters are described in terms of the input parameters. Built from the Design of Experiments (DOE), they quickly provide the approximated values of the output parameters throughout the design space without having to perform a complete solution. The accuracy of a response surface depends on several factors: the complexity of the variations of the solution, the number of points in the original DOE, and the response surface type. Once a response surface has been generated, you can create and manage response points and charts. These postprocessing tools help you to understand how each output parameter is driven by input parameters and how you can modify your design to improve its performance.



Clicking the ellipsis button [...] by the View for Response Surface opens a dialog that lets you adjust the variables selected and the values applied.



Tuning a Response Surface

For the X Axis and Y Axis, you can specify a Resolution, and the variable to use. For the Z Axis you can select the Cost or calculation. For variables not selected for the X and Y axis, a slider is enabled that lets you adjust the value to see the effect on the response surface plot. You can enable or disable Real time mode by using the check box at the lower left.

Full Range Axis check box

When it is checked all Axes are set to their maximum ranges, and the ranges won't be changed while tuning unless you change the axis variable.

When it is unchecked, the Y(2D)/Z(3D) axis range is auto updated to fit the curve/surface.

Accumulate Response Curve

It always be unchecked when the tuning dialog started, and won't restore it last check/uncheck state. When it is checked, it will retaining the existing curves, and add new curve to the plot. Uncheck the checkbox won't clear the accumulated curves, just stop to accumulate new curve. When Axis variable is changed all accumulated curves will be cleared.

Exporting Response Surface Data

You can Export the Table in the following formats:

- Comma delimited data files, (*.csv)
- Tab delimited data files, (*.tab)

- Ansoft Plot data files, (*.dat)
- Post Processor format data files, (*.txt)

Related Topics

Setting Up Design of Experiments

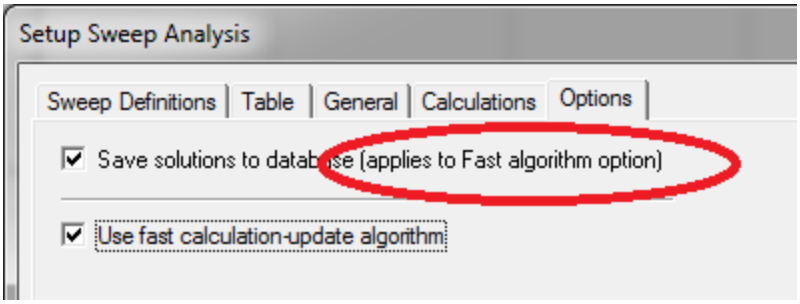
Using Design of Experiments

Using the Fast Calculation-Update Algorithm

A fast calculation-update algorithm is available to speed up Optimetrics and report updates during Optimetrics analyses. The fast calculation-update algorithm will generate the same Optimetrics results, only faster, and is available for all Optimetrics analyses *except* Tuning analysis. By default, the fast calculation-update option is automatically enabled whenever it is applicable, but you can configure it manually using the **Options** tab of the Optimetrics **Setup** dialog box. To enable the fast calculation-update algorithm:

1. Click your product, and then **>Optimetrics Analysis >Add** <*OptimetricsType*>.

   A typical dialog to **SetupSweep Analysis** is shown below with the **Options** tab selected.

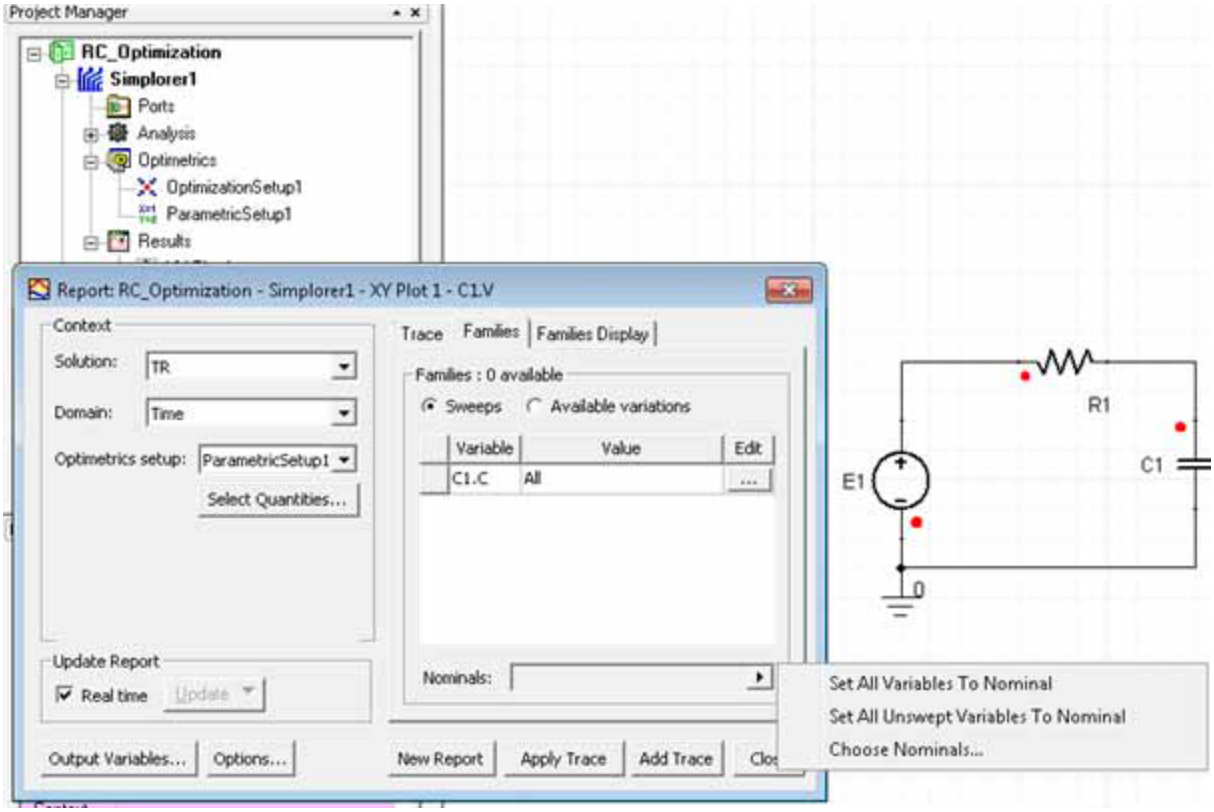


   The **Options** tab is the same for all optimization setups.

2. Check **Use fast calculation-update algorithm** to enable use of the algorithm. (See also, *Fast Calculation-Update Algorithm Limitations*.)

   When the fast calculation-update algorithm is enabled:

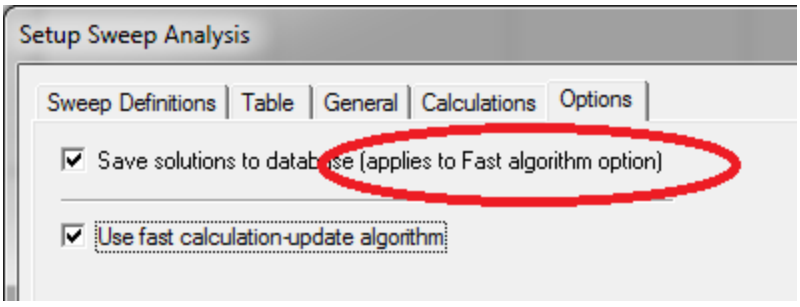
   - You can enable some reports to be updated automatically during the optimetrics analysis if you **Set All Variables To Nominal** in the Report/Trace setup dialog:



- You can see each trace (overwriting the previous), by setting the **Optimetrics setup** in the Report dialog to **None** in addition to having all variables set to Nominal. At the end of the analysis, the user will see the last calculated value.

3. If you have enabled **Use fast calculation-update algorithm** and want to save the solution data for every solved design variation in the Optimetrics analysis, select **Save solutions to database** as shown below. Selecting **Save solutions to database** has no effect without enabling **Use fast calculation-update algorithm**.

Note	Do not select this option when requesting a large number of iterations as the data generated will be very large and the system may become slow due to the large I/O requirements.



- When the **Save solutions to database** option is checked, a plot with traces based on the Optimetrics Setup just run can be updated through a menu command (right-click the desired report in the Project Manager Results folder and select **Update Report**) and will show results as appropriate (family, if chosen).
- Without the **Save solutions to database** option checked, you can examine analysis data in the **Post Analysis Display** dialog box, available by right-clicking on the Optimetrics Setup and choosing **View Analysis Result.**

Fast Calculation-Update Algorithm Limitations

The fast calculation-update algorithm **cannot** be used if **any** optimetrics calculation uses:

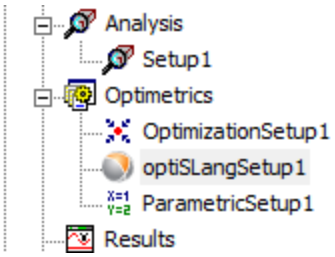
- **Project/Design variables** – If the project/design variable is ***not*** swept in the Optimetrics analysis, and you would like to use it in the expression: You can create an output variable for the Project/Design variable, assign the value of the Project/Design variable to the output variable, and use the output variable in the expression instead.
- **More than one range function** – For example, when range function is not the outermost function or when range function takes arguments.
- **More than one calculation range**, or the calculation range is not for primary sweep.

Similarly, when the fast calculation-update algorithm is enabled, a trace cannot be updated during an analysis if the trace expression uses:

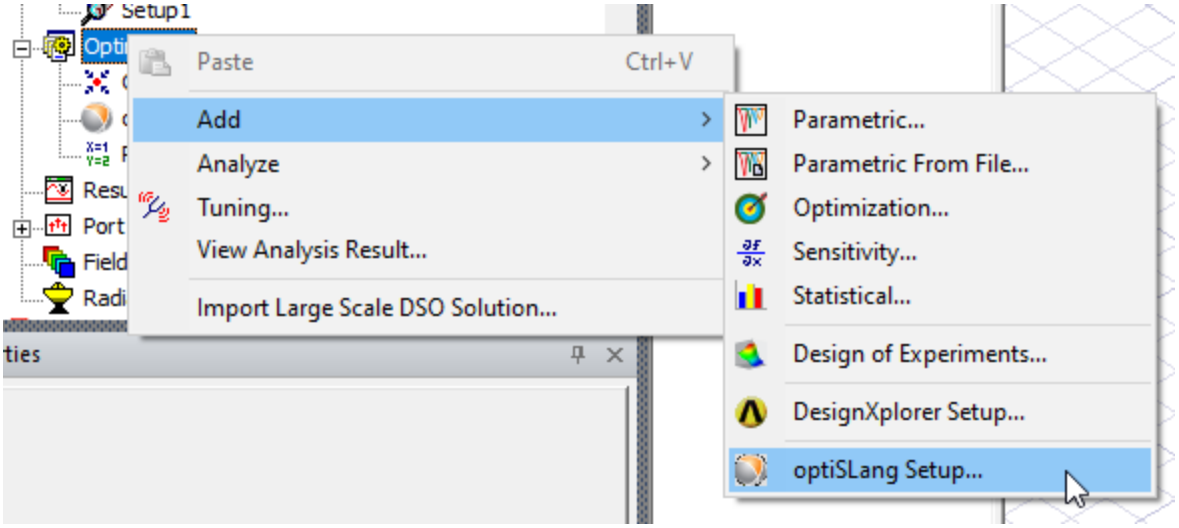
- **Project/Design variables**. (However, you can use the same workaround described above.)
- **Any range function.**

Setting Up optiSLang in Ansys Electronics Desktop

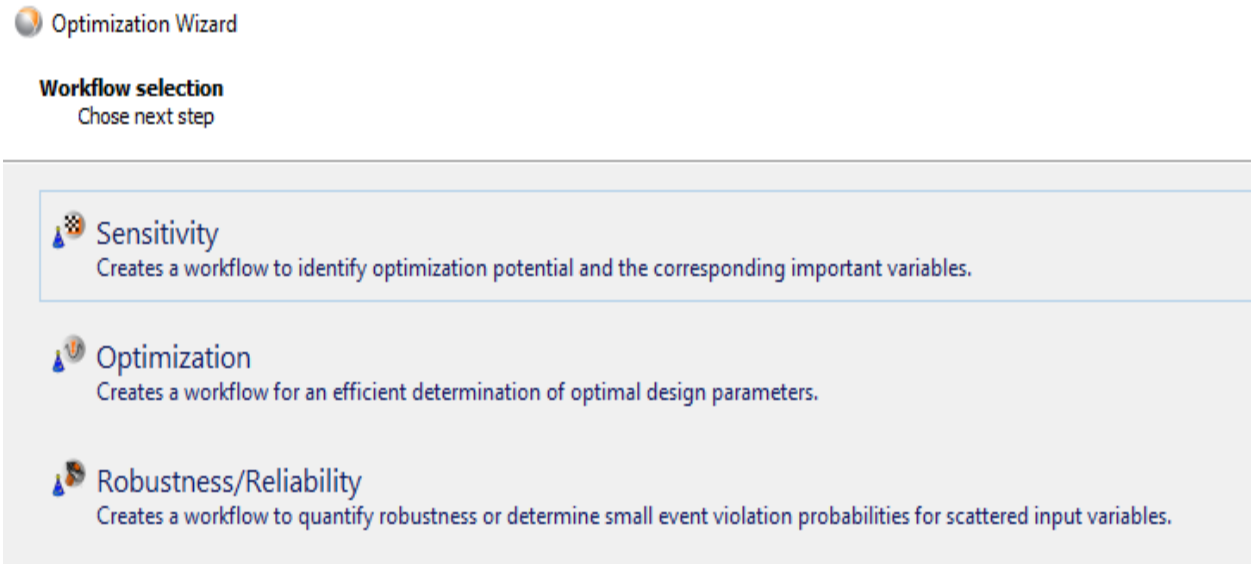
optisLang is a tool that use for graphical programming, process integration and automation that can be integrated with Ansys Electronics Desktop for optimization analysis. It is available at the Ansys Customer Portal as a Downloaded Add-on package. If you have installed optiSLang, and enabled the feature for Ansys Electronics Desktop, you create an optiSLang setup in much the way you would create an Optimetrics setup.



Ansys optiSLang integration then lets you create an optiSLang project from a setup you define in Ansys Electronics Desktop Optimization.



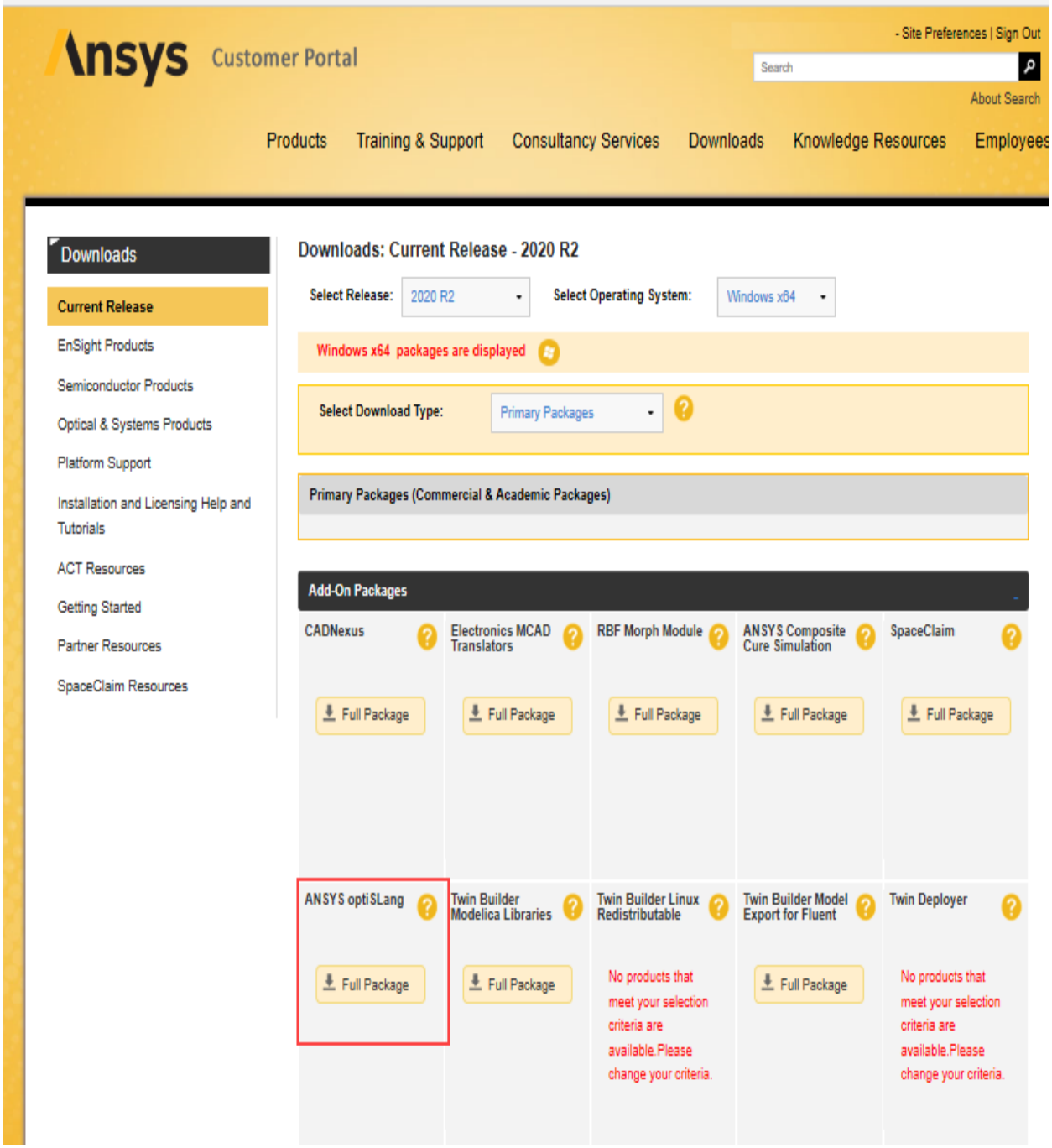
This opens an optiSLang Optimization wizard from which you can define a simulation.



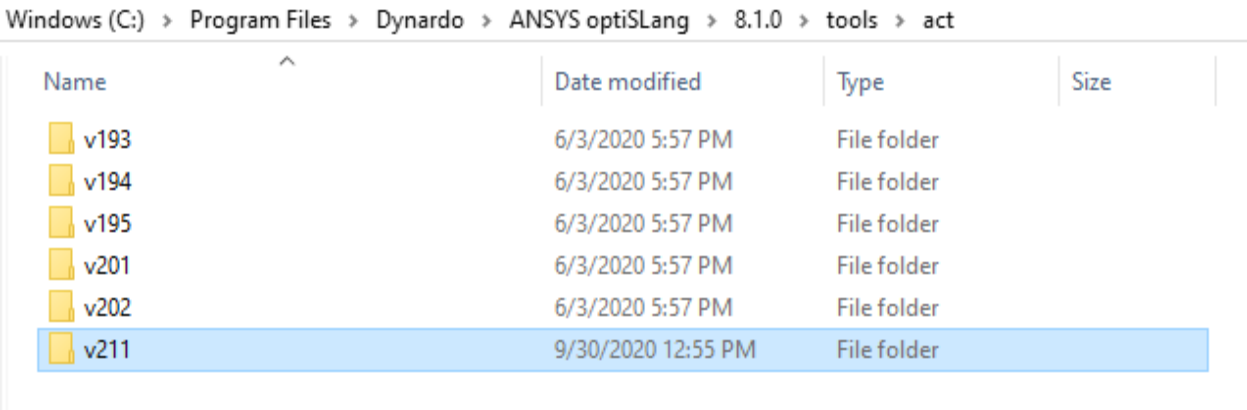
The process for setting up and using optisLang from Ansys Electronics Desktop follows.

Prerequisites

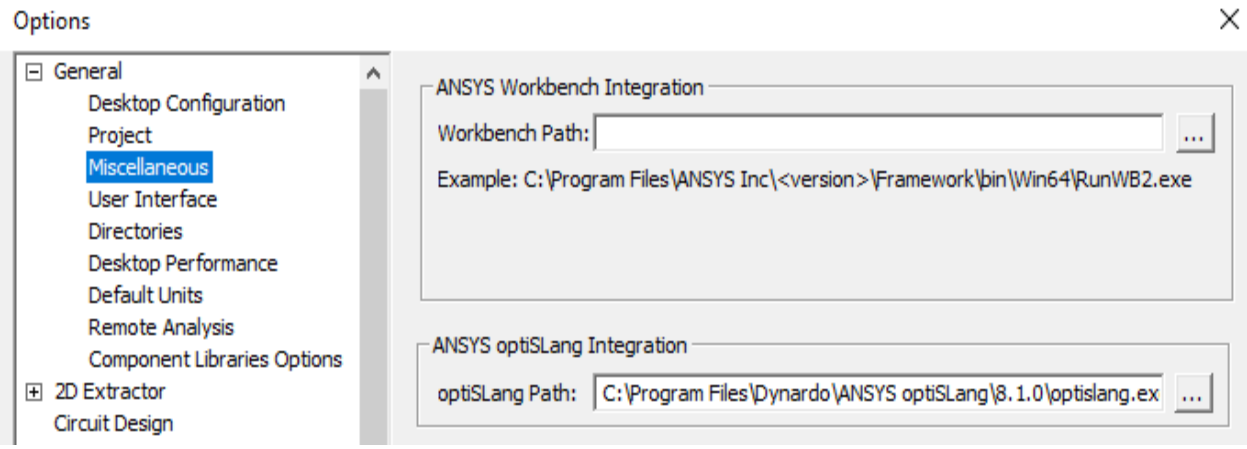
You must obtain optiSLang 8.1.0 or higher from the Ansys Customer Portal as a Downloaded Add-on package and install locally.



You must also install the optiSLang extensions to the scripting folder of the optiSLang installation.

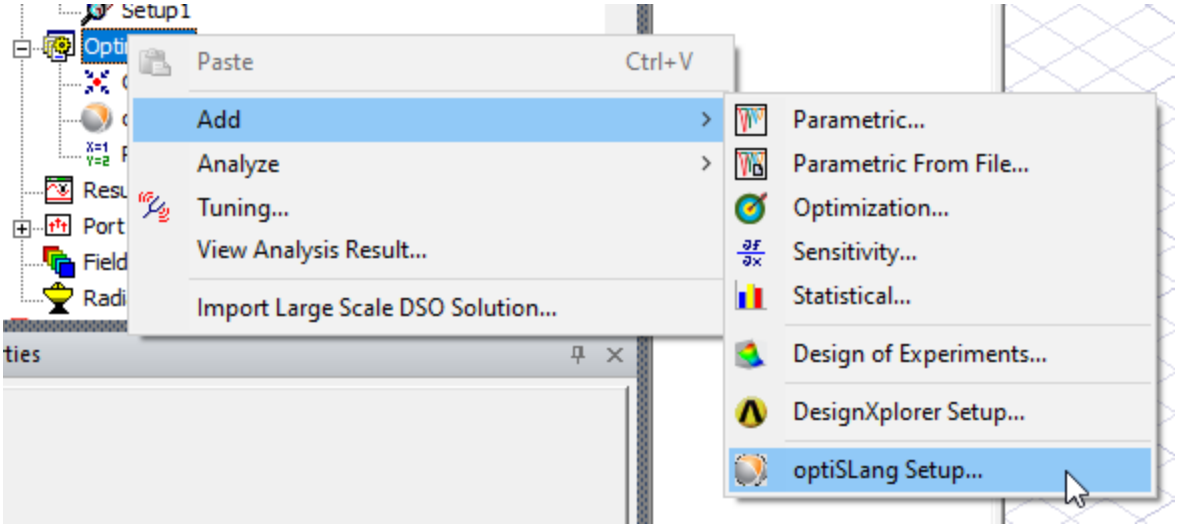


In the **Tools>General Options>Miscellaneous** options, specify the path to the optiSLang installation.

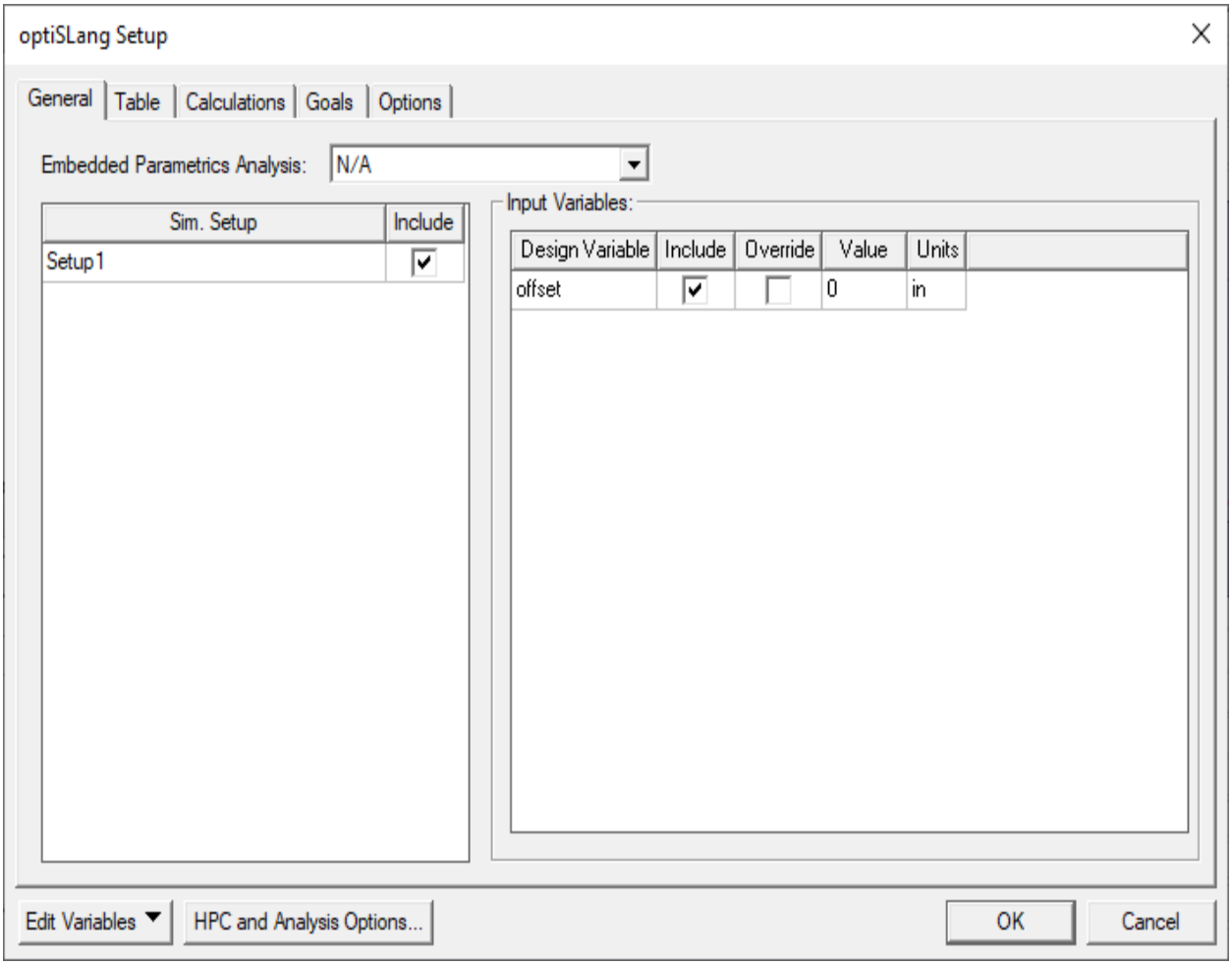


Add the optiSLang Setup

Right-click the Optimetrics icon in the Project tree and select **Add> optiSLang Setup...**. You can also use **Maxwell 2D (or Maxwell 3D)> Optimetrics Analysis> Add optiSLang Setup...**.

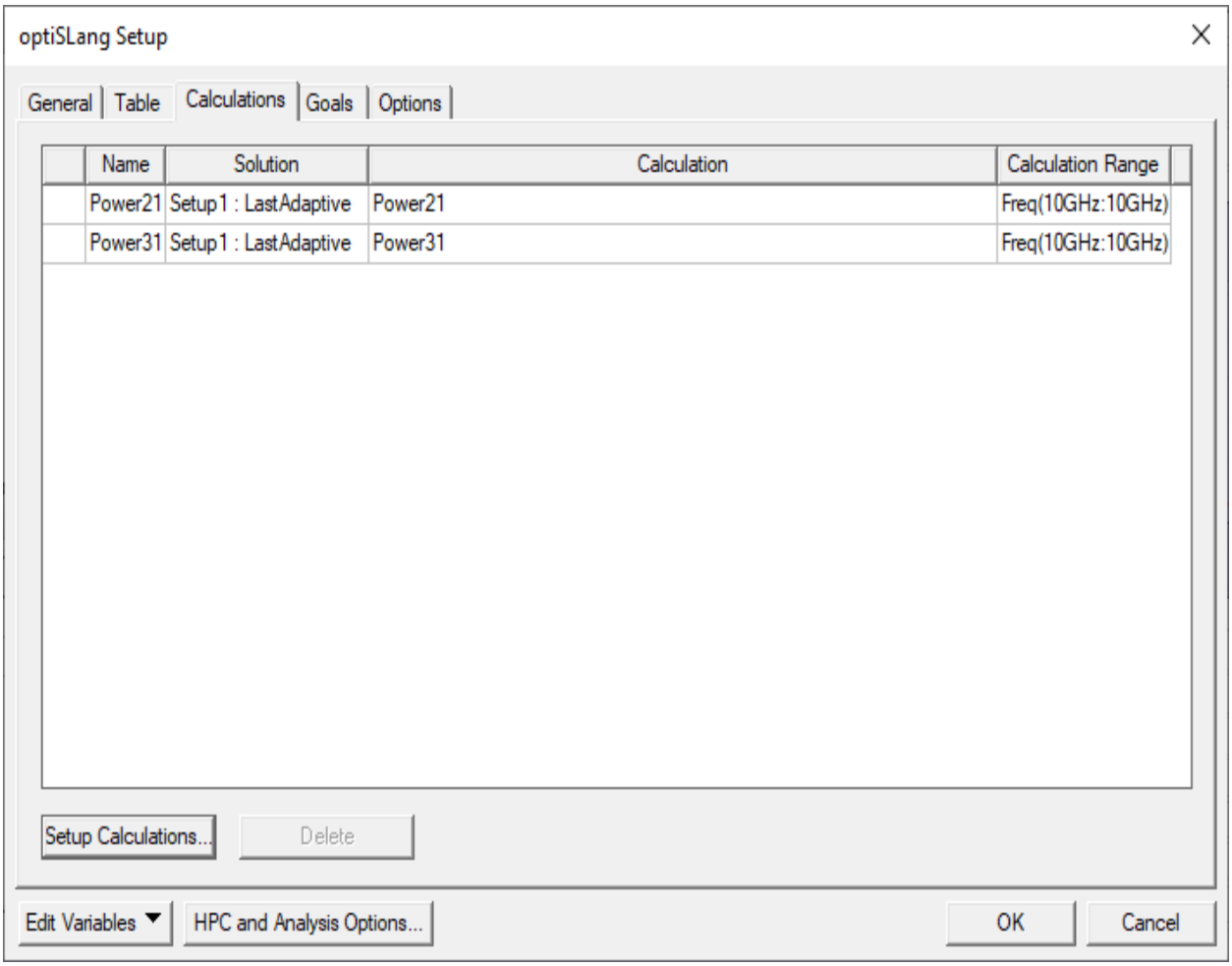


The *optiSLang Setup* dialog box appears. It is very similar to an Optimetrics setup.



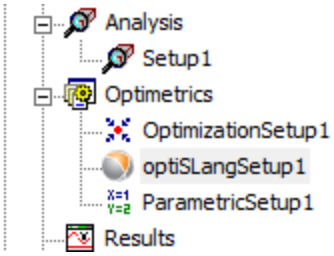
It has tabs for:

- General, which specifies the Setup to include and the Design Variables to use, and whether to include and/or Override the value.
- Table, which shows the values of the included variable.
- Calculations, which accesses Optimetrics calculations.



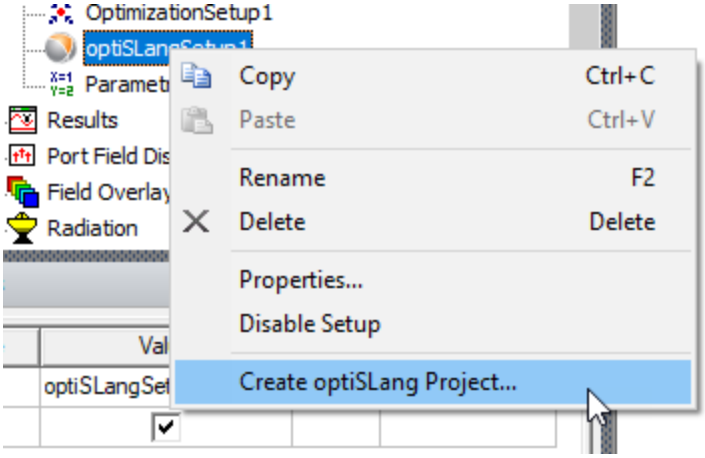
- Goals, including cost functions calculations and norm type.
- Variables, previously defined for Optimization/Design of Experiments as Project or Design variables, and whether to include them, treat as discrete, whether to Use Manufacturable Variables, and the available Levels.
- Options, to Save Fields and Mesh, and whether to copy geometrically equivalent meshes.

Once you have set parameters and click OK, the optiSLang setup appears under the Optimetrics icon in the Project tree.



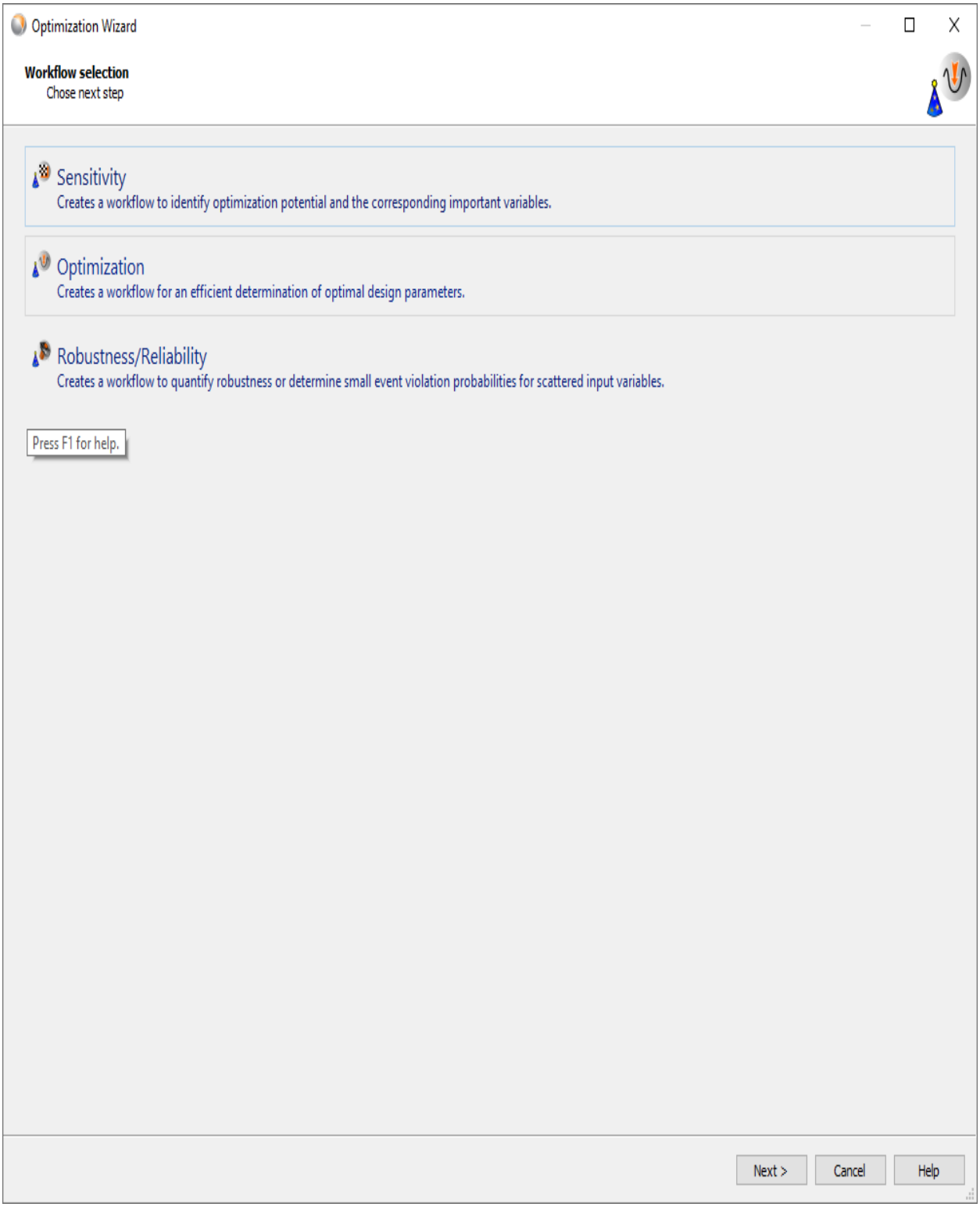
Creating an optiSLang Project

From here you can right-click the optiSLang setup for the shortcut menu.

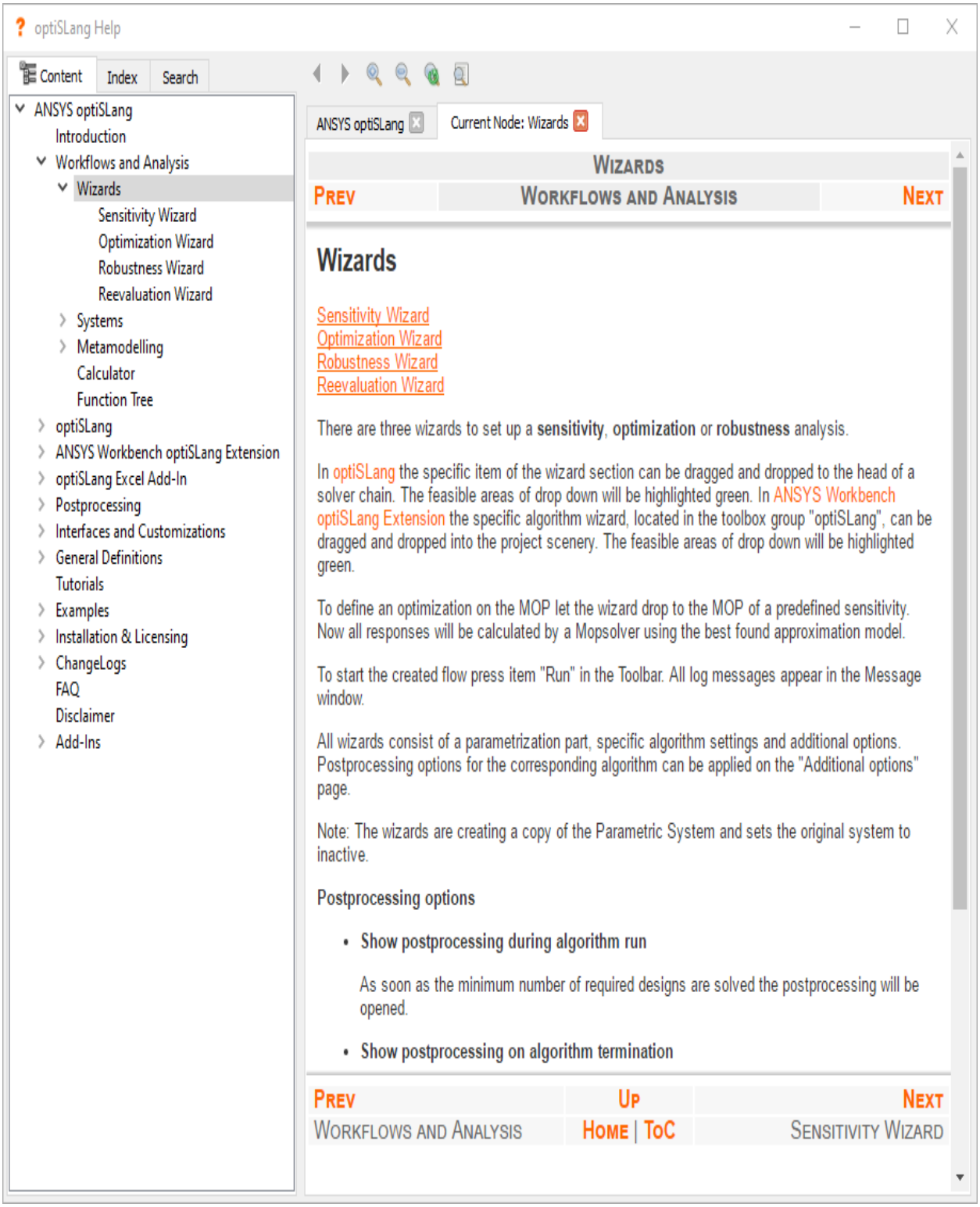


When you select **Create optiSLang Project...**, a progress window opens.

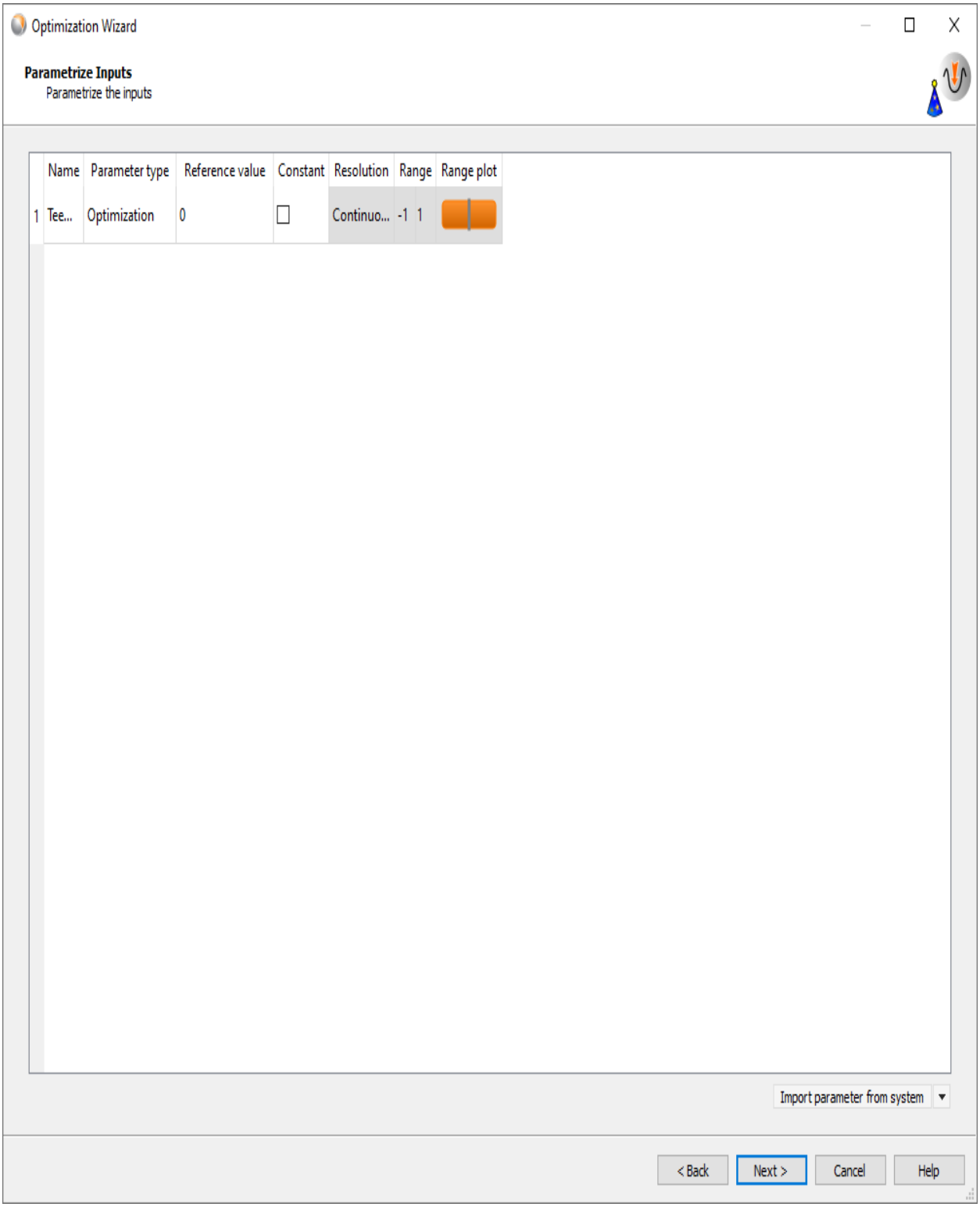
During this progress, the optiSLang **Optimization wizard** displays, where you can select the Workflow as Sensitivity, Optimization, or Robustness/Reliability.



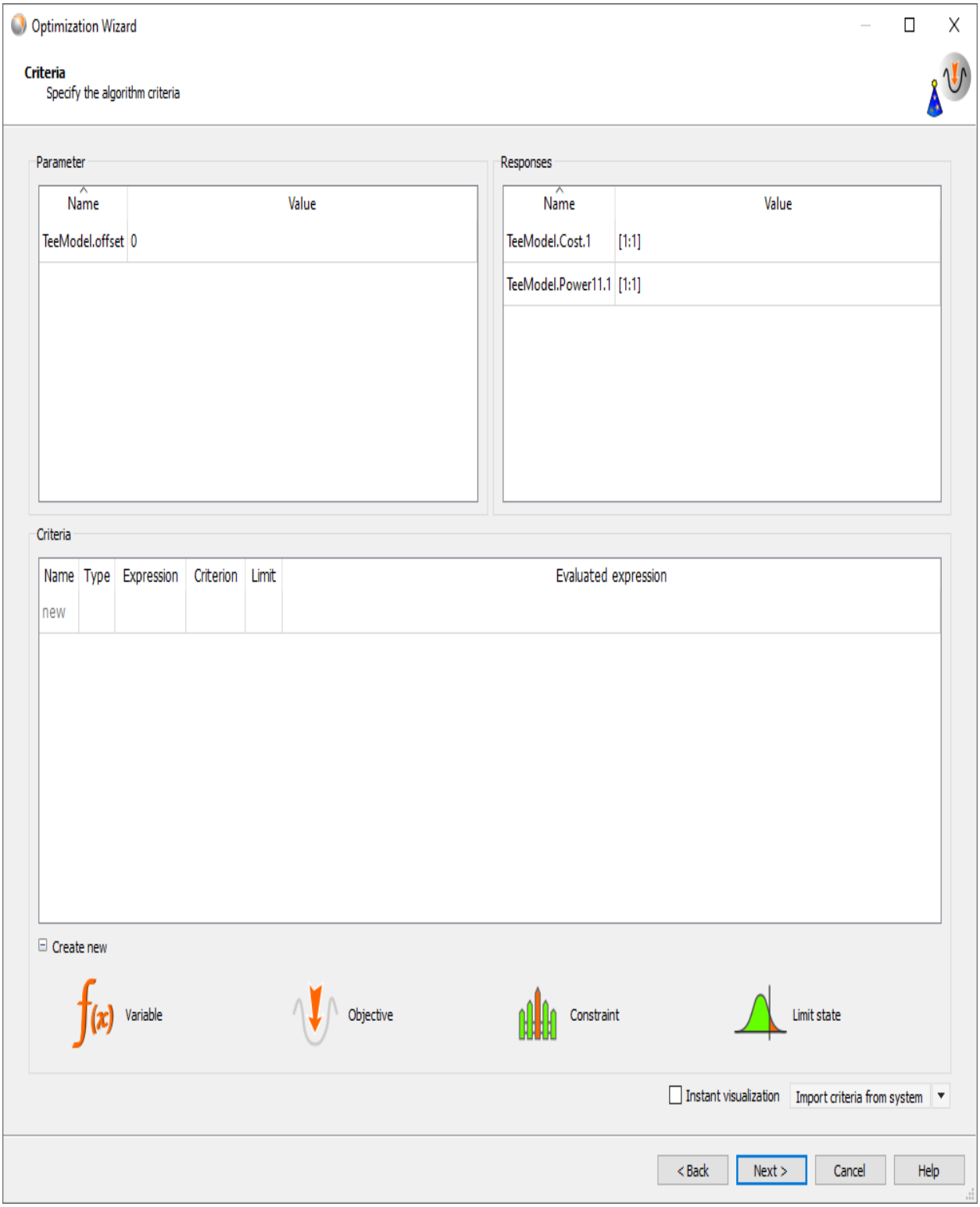
The **Help** button opens a separate **optiSLang Help** window.



Selecting a Workflow and clicking **Next** opens a **Parameterize Input**s page.



Select **Next** for the **Criteria** page.



Related Topics:

Optimetrics

Tuning Overview

Tuning a variable is useful when you want to manually modify its value and immediately perform an analysis of the design. For example, it is useful after performing an optimization analysis, in which Optimetrics has determined an optimal variable value, and you want to fine tune the value to see how the design results (for example, traces in a report) are affected.

A design can be updated after a tuning analysis to reflect a design variation solved during a tuning analysis and the results, including field solutions if you select **Save Fields and Mesh on the Options tab of the setup**.

Related Topics

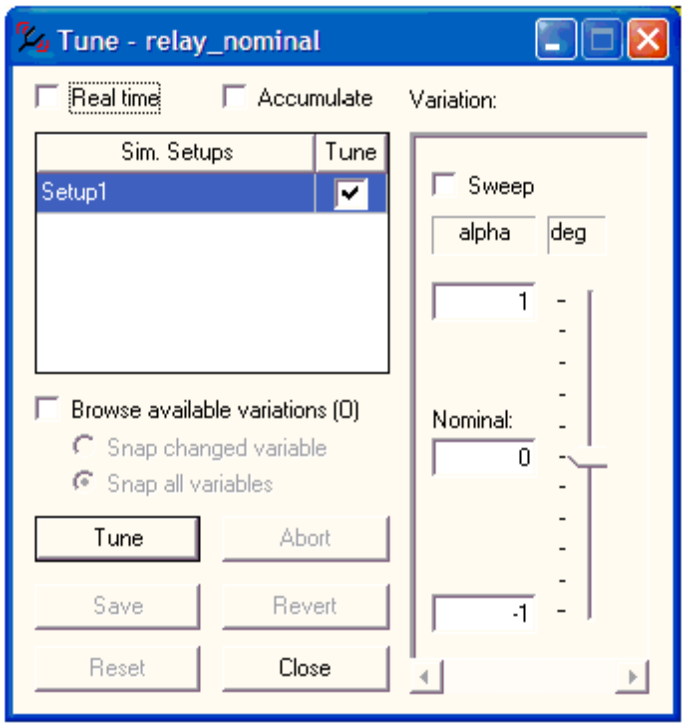
Tuning a Variable

Tuning a Variable

If you want to ensure that tuning does not resolve variations already solved by an optimization setup, you must check **Save Fields and Mesh** in the **Options** tab of that setup.

1. Before a variable can be tuned, you must specify that you intend for it to be used during a tuning analysis in a Project or Design **Properties** dialog box.
2. On the **Maxwell3D or Maxwell2D** menu, click **Optimetrics Analysis>Tune**  .

   The **Tune** dialog box appears, listing the variables which have been included for tuning.



3. Clear the **Real Time** option.

   Clearing the **Real Time** option enables the **Tune** button. If this option is selected, a simulation begins immediately after you move the slider. Otherwise, you use the **Tune** button to apply the current values to a simulation.

4. If you want to see updates to an open Report plot while tuning a post processing variable, you must select the **Browse available variations** check box. Selecting **Browse available variations** disables the sweep check box, and the fields for minimum and maximum variable values. This feature lets you see the effect of changes to the post processing variables on plotted results.

   Clearing **Browse available variations** enables the Sweep check box, the minimum and maximum fields, and changes the Nominal field to Step. See step 6.

5. In the **Sim. Setups** column, select the solution setup you want Maxwell to use when it solves the specified design variation.

   Maxwell solves the analysis using the solution setup you select. If you select more than one, results are generated for all selected solution setups.

   Checking the Tune box for a Sim Setup enables the Real Time check box, the Browse available variations check box, and the Snap radio buttons. Clearing the Tune box disables those selections.

6. In the **Nominal** text box for the variable you want to tune, type the value of the variable you want Maxwell to solve, or drag the slider to increase or decrease its value.

Warning	Variable values must be single real numbers, or expressions that evaluate to single real numbers. Complex numbers cannot be used as the values of variables in any optimetric analysis.

Alternatively, if you want Maxwell to solve a range of values, specify a linear range of values with a constant step size:

a. Select the **Sweep check box**.
b. In the text box below the **Step** value, type the starting value in the variable range.
c. Type the step size, or difference between variable values in the sweep definition, in the **Step** text box. The step size determines the number of design variations between the start and stop values. Maxwell solves the model at each step in the specified range, including the start and stop values.
d. In the text box just below the variable name, type a stopping value in the variable range.

7. If you have cleared the Real Time check box, click **Tune** to apply the changes you have made to the variable values.

Note	Sweeping or using a complex variable is not allowed in any optimetrics setup, including optimization, statistical, sensitivity, and tuning setups.

8. Changing a variable value with the sliders of by typing in the text field enables the **Save** and **Reset** buttons.

   Clicking **Save** opens a **Save As** dialog with a name field, and an **Apply tuned values to design** check box.

   Clicking **Reset** changes the variable values back to what they were originally.

9. If you have changed one or more included variables, clicking **Close** on the **Tuning** dialog opens the **Apply Tuned Variation** dialog box. This lists the included variables and the values for each tuning. If you have tried multiple values, they are listed, and the current value is highlighted. Select another value to change the highlight. Click **OK** to apply the highlighted values to the design, or **Don't Apply** to ignore the changes from the original variable values.

   If you have applied variant values, you should see the new values listed in the relevant Design or Project Properties lists of variables and values, and if the changes affect plots or physical features of a model, those changes should also appear.

   Click **Cancel** to close the dialog and go back to the **Tune** dialog.

Related Topics

Applying a Tuned State to a Design
Tuning Overview
Resetting Variable Values after Tuning

Applying a Tuned State to a Design

You can apply the variable values solved during a tuning analysis to the nominal design in one of the following three ways:

- When closing the **Tune** dialog box:

1. Click **Close** to exit the **Tune** dialog box.

   The **Apply Tuned Variation** dialog box appears.

2. Click the design variation you want to apply, and then click **OK**.

   The variable values from the solved design variation become the current variable values for the nominal design. If you have applied variant values, you should see the new values listed in the relevant Design or Project Properties lists of variables and values, and if the changes affect plots or physical features of a model, those changes should also be apparent.

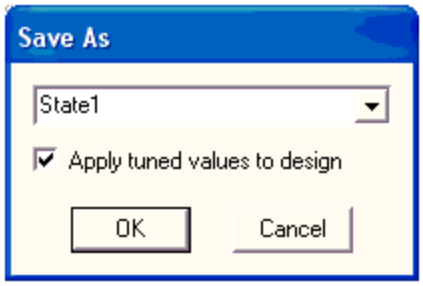
   - When saving a tuned state.
   - When reverting to a tuned state.

Saving a Tuned State

You can save the settings in the **Tune** dialog box, including the variable values you specified for a tuning analysis. Saved states are only available during the current session of the **Tune** dialog box; they are not stored for the next session.

1. After tuning a variable, click **Save** in the **Tune** dialog box.

   A **Save As** dialog box appears.



2. Type a name for the tuned state in the text box.
3. Select **Apply tuned values to design** if you want to update the model to the new variable values.
4. Click **OK** to return to the **Tune** dialog box.

Related Topics

Reverting to a Saved Tuned State

Reverting to a Saved Tuned State

You can revert to a group of saved settings in the **Tune** dialog box, including the variable values you specified for a specific tuning analysis. Saved states are only available during the current session of the **Tune** dialog box; they are not stored for the next session.

1. In the **Tune** dialog box, click **Revert**.

   The **Revert** dialog box appears.
2. Type the name of the tuned state you want to apply or click a name in the pull-down list.
3. Select **Apply tuned values to design** if you want to update the model to the selected tuned state's variable values.
4. Click **OK** to return to the **Tune** dialog box.

Related Topics

Saving a Tuned State

Resetting Variable Values after Tuning

If you want to reset variable values to the values they were set to when you started the current session of the **Tune** dialog box:

- After tuning a variable, click **Reset** in the **Tune** dialog box.

  Solutions for the design variations solved during tuning analyses remain available for post processing.

Saving Field Solutions for Optimetrics Analyses

In order to preserve disk space, by default Maxwell does not save field solution data for every solved design variation in an optimization analysis. It only saves the field solutions for the nominal design when an adaptive analysis is specified in the solution setup or when you request that fields be saved for each solved point in a frequency sweep. If the nominal design is not included in the optimization analysis, all field solutions are deleted.

To save the fields for all design variations, change the default setting for all projects:

1. Select **Tools>Options**, and then select either **Maxwell3D Options or Maxwell2DOption**.

   The appropriate **Options** dialog box appears.
2. Under the **General** tab, select **Save Optimetrics field solutions**.

   Save Fields is selected by default when you create a new Optimetrics setup.

Related Topics

Saving Field Solutions for a Parametric Setup

Saving Field Solutions for an Optimization Setup

Saving Field Solutions for a Sensitivity Setup

Saving Field Solutions for a Tuning Analysis
Saving Field Solutions for a Statistical Setup
Copy Geometrically Equivalent Meshes

Saving Field Solutions for a Parametric Setup

In order to preserve disk space, by default Maxwell does not save field solution data for every solved design variation in a parametric setup. It only saves the field solutions for the nominal design. If the nominal design is not included in the parametric setup, by default field solutions will not be available.

To save the fields for all design variations solved during a parametric analysis:

1. Either **Add Sweep** or right click on an existing sweep to open the **Setup Sweep Analysis** dialog box.
2. Select the **Options** tab.
3. Click the **Save Fields And Mesh check box**.Optionally, select **Copy geometrically equivalent meshes** and choose whether to **Solve with copied meshes only**, or to **Solve with copied meshes and continue adaptive passes**

   Maxwell will save the field solution data for every solved design variation in the parametric setup.

Related Topics

Saving Field Solutions for Optimetrics Analyses

Saving Field Solutions for an Optimization Setup

In order to preserve disk space, by default Maxwell does not save field solution data for every solved design variation in an optimization analysis. It only saves the field solutions for the nominal design when an adaptive analysis is specified in the solution setup or when you request that fields be saved for each solved point in a frequency sweep. If the nominal design is not included in the optimization analysis, all field solutions are deleted.

To save the fields for all design variations solved during an optimization analysis:

1. Open an **Edit Sweep** dialog by either adding a sweep or right-click on a an existing sweep to view the short cut menu and selecting Properties.
2. Select the **Options** tab.
3. Click the **Save Fields And Mesh check box**.Optionally, select **Copy geometrically equivalent meshes** and choose whether to **Solve with copied meshes only**, or to **Solve with copied meshes and continue adaptive passes**

   Maxwell will save the field solution data for every solved design variation in the optimization setup.

Related Topics

Saving Field Solutions for Optimetrics Analyses

Saving Field Solutions for a Sensitivity Setup

In order to preserve disk space, by default Maxwell does not save field solution data for every solved design variation in a sensitivity analysis. It only saves the field solutions for the nominal design when an adaptive analysis is specified in the solution setup or when you request that fields be saved for each solved point in a frequency sweep. If the nominal design is not included in the sensitivity analysis, all field solutions are deleted.

To save the fields for all design variations solved during a sensitivity analysis:

1. Open the **Setup Sensitivity Analysis** dialog box.
2. Select the **Options** tab.
3. Click the **Save Fields And Mesh check box**. Optionally, select **Copy geometrically equivalent meshes** and choose whether to **Solve with copied meshes only**, or to **Solve with copied meshes and continue adaptive passes**

   Maxwell will save the field solution data for every solved design variation in the sensitivity analysis.

Related Topics

Saving Field Solutions for Optimetrics Analyses

Saving Field Solutions for a Tuning Analysis

In order to preserve disk space, by default Maxwell does not save field solution data for every design variation solved in a tuning analysis. It only saves the field solutions for the nominal design when an adaptive analysis is specified in the solution setup or when you request that fields be saved for each solved point in a frequency sweep. If the nominal design is not included in the tuning analysis, all field solutions are deleted.

To save the fields for all design variations solved during a tuning analysis:

- In the **Tuning** dialog box, select **Save Fields**.

  Maxwell will save the field solution data for every solved design variation in a tuning analysis.

Related Topics

Saving Field Solutions for Optimetrics Analyses

Saving Field Solutions for a Statistical Setup

In order to preserve disk space, by default Maxwell does not save field solution data for every design variation solved in a statistical analysis. It only saves the field solutions for the nominal design when an adaptive analysis is specified in the solution setup or when you request that fields be saved for each solved point in a frequency sweep. If the nominal design is not included in the statistical analysis, all field solutions are deleted.

To save the fields for all design variations solved during a statistical analysis:

1. Open the **Setup Statistical Analysis** dialog box.
2. Select the **Options** tab.
3. Click the **Save Fields And Mesh check box**. Optionally, select **Copy geometrically equivalent meshes** and choose whether to **Solve with copied meshes only**, or to **Solve with copied meshes and continue adaptive passes**

   Maxwell will save the field solution data for every solved design variation in the statistical setup.

Related Topics

Saving Field Solutions for Optimetrics Analyses

Saving Fields Solutions for a Design of Experiments Setup

In order to preserve disk space, by default Maxwell does not save field solution data for every solved design variation in a Design of Experiments setup. It only saves the field solutions for the nominal design. If the nominal design is not included in the setup, by default field solutions will not be available.

To save the fields for all design variations solved during a Design of Experiments analysis:

1. Open the **Design of Experiments** dialog box.
2. Select the **Options** tab.
3. Click the **Save Fields And Mesh** check box. Optionally, select **Copy geometrically equivalent meshes** and choose whether to **Solve with copied meshes only**, or to **Solve with copied meshes and continue adaptive passes**.

   Maxwell will save the field solution data for every solved design variation in the Design of Experiments setup.

Related Topics

Saving Field Solutions for Optimetrics Analyses

Copying Meshes in Optimetrics Sweeps

An option in the Optimetrics Analysis setups allows you to request Maxwell to copy a mesh that was calculated for one sweep variation for reuse on a geometrically-equivalent sweep variation. For example, with this option selected a sweep on a scan angle would not need to generate meshes for each solution. The option is available on the setups for sweeps on parametrics, optimization, sensitivity, and statistics.

To copy and reuse meshes on geometrically-equivalent parametric variations:

1. Define a variable for the kind of Optimetrics sweep you intent to setup.
2. Select **Maxwell3D or Maxwell2D** and then select the appropriate **Optimetrics>Add** command to display a **Setup** dialog box.
3. Click the **Options** tab in the Setup dialog box.
4. Select **Copy geometrically equivalent meshes**.

   Maxwell will copy the mesh solution calculated for a particular parametric sweep for reuse on each geometrically-equivalent sweep variation.

Note	This option is available with all Optimetrics setups, and is applied when these analyses generate geometrically-equivalent values. However, it is most relevant to parametric sweep, where such equivalences are more likely to occur.

The **Copy geometrically equivalent mesh** option is not recommended for use when the frequency is varying, since meshing is frequency-dependent. You may wish to turn this option off when the first geometrically equivalent variation requires numerous passes after the initial mesh, but the other geometrically-equivalent variations require fewer additional passes, so that it is cheaper to start with the initial mesh each time.

Adding an Expression in the Output Variables Window

When you are in the **Output Variables** window (after clicking **Edit Calculation** from the one of the setup analysis windows), do the following to specify an expression:

1. Type a name for the expression in the **Name** text box.
2. Do the following in the **Calculation** section of the window to insert a quantity into the expression:
   a. Select the **Report Type** and **Solution** from the pull-down lists.
   b. Select a **Category**, **Quantity**, and **Function** from the lists, and click **Insert Quantity Into Expression**.
   c. If you want to insert a specific pre-defined function, select one from the **Function** pull-down list, and click **Insert Function**.
3. You can also type numbers or expression by hand directly into the **Expression** area.

Excluding a Variable from an Optimetrics Analysis

To exclude a variable from being optimized or included in a sensitivity or statistical analysis:

1. Do one of the following:
   - In the **Setup Optimization** dialog box, click the **Variables** tab.
   - In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.
   - In the **Setup Statistical Analysis** dialog box, click the **Variables** tab.

   All of the independent variables that were selected for the optimization analysis are listed.
2. Clear the **Include** option for the variable you want to exclude from the analysis.

   The **Override** option is now selected. This indicates that, for this optimization analysis, the variable is not included.

Note	Alternatively, you can select the **Override** option first, and then clear the **Include** option for the variable you want to exclude.

3. Click **OK**.

Modifying the Value of a Fixed Variable

If you are not including a variable in an optimization, sensitivity, or statistical analysis, Optimetrics uses that variable's current value during the analysis.

To override the current value of a fixed variable for an Optimetrics setup:

1. Do one of the following:
   - In the **Setup Optimization** dialog box, click the **Variables** tab.
   - In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.
   - In the **Setup Statistical Analysis** dialog box, click the **Variables** tab.
2. Click **Set Fixed Variables**.

   The **Setup Fixed Variables** dialog box appears. Under **Fixed Variables**, all of the current independent variable values are listed.
3. Click the **Value** text box of the variable with the value you want to override.
4. Type a new value in the **Value** text box, and then press **Enter**.

   The **Override** option is now selected. This indicates that the value you entered is used for this Optimetrics setup; the current variable value set for the nominal design is ignored.

Note	Alternatively, you can select the **Override** option first, and then type a new value in the **Value** text box.

5. Optionally, click a new unit system in the **Units** text box.
6. Click **OK**.

To revert to a default variable value, clear the **Override** option.

Linear Constraints

Once the optimization variables are specified, the optimizer handles each of them as an *n*-dimensional vector *x*. Any point in the design space corresponds to a particular *x*-vector and to a design instance. Each design instance may be evaluated via Finite Element Analysis and assigned a cost value; therefore, the cost function is defined over the design space (cos*t*(*x*): $R^n \rightarrow R$), where *n* is the number of optimization variables.

In practice, a solution of the minimization problem is sought only on a bounded subset of the R^n space. This subset is called the feasible domain and is defined via linear constraints.

You may constrain the feasible domain of a design variable by defining linear constraints for the optimization process. The feasible domain is defined as the domain of all design variables that satisfy all upper and lower bounds and constraints. Linear constraints are defined by the following inequalities:

$$\sum_i \alpha_{ij} x_i < c_j \forall j$$

where

- α_{ij} are coefficients.
- c_j is a comparison value for the j^{th} linear constraint.
- x_i is the i^{th} designer parameter.

Related Topics

Setting a Linear Constraint

Setting a Linear Constraint

A linear constraint defines the linear relationship between variables. Setting linear constraints in Optimetrics is useful for establishing limitations involving linear combinations of variable values.

1. Do one of the following:
   - If you are setting up an optimization analysis: In the **Setup Optimization** dialog box, click the **Variables** tab.
   - If you are setting up a sensitivity analysis: In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.
2. Click **Linear Constraint**.

   The **Linear Constraint** dialog box appears.
3. Click **Add**.

   The **Edit Linear Constraint** dialog box appears.
4. Click a **Coeff** text box and type a positive or negative coefficient value.
5. Click a condition, **<** (less than) or **>** (greater than), from the pull-down list.

6. Type the inequality value, which should be a constant value, in the text box to the right of the condition.
7. Click **OK**.

   You return to the **Linear Constraint** dialog box. The left-hand side of the constraint appears in the **LHS** (left-hand side) column. The condition is listed in the **Condition** column, and the inequality value is listed in the **RHS** (right-hand side) column.

Related Topics

Modifying a Linear Constraint

Deleting a Linear Constraint

Linear Constraints

Modifying a Linear Constraint

1. Do one of the following:
   - If you are setting up an optimization analysis: In the **Setup Optimization** dialog box, click the **Variables** tab.
   - If you are setting up a sensitivity analysis: In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.
2. Click **Linear Constraint**.

   The **Linear Constraint** dialog box appears.
3. Click the row listing the constraint you want to modify, and then click **Edit**.

   The **Edit Linear Constraint** dialog box appears.
4. Optionally, click a **Coeff** text box and type a new coefficient value.
5. Optionally, click a different condition, **<** (less than) or **>** (greater than), in the pull-down list.
6. Optionally, type a different inequality value in the text box to the right of the condition, and then click **OK**.

   You return to the **Linear Constraint** dialog box. The new coefficient value, the condition, and the inequality value appear in the **LHS** (left-hand side), **Condition**, and **RHS** (right-hand side) columns, respectively.

Deleting a Linear Constraint

1. Do one of the following:
   - If you are setting up an optimization analysis: In the **Setup Optimization** dialog box, click the **Variables** tab.
   - If you are setting up a sensitivity analysis: In the **Setup Sensitivity Analysis** dialog box, click the **Variables** tab.
2. Click **Linear Constraint**.

   The **Linear Constraint** dialog box appears.

3. Click the row listing the constraint you want to delete, and then click **Delete**.

   The constraint is deleted.

Running an Optimetrics Analysis

Once you have created all necessary Optimetrics based analyses, you have several options for running the simulations.

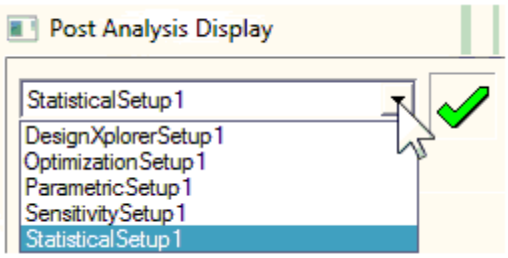
- To use the **Analyze All** command at the Project or design level to simulate the nominal problem and subsequently run all Optimetrics setups, do the following:
  1. In the Project Manager window, right-click on the **project** or **design** name.
  2. Click **Analyze All** from the shortcut menu.
- To use the **Analyze All** command from the Optimetrics menu to simulate only the Optimetrics based setups, do the following:
  1. In the Project Manager window, right-click on **Optimetrics**.
  2. Click **Analyze>All** from the shortcut menu.
- You can choose to analyze only the setups related to a specific Optimetrics type of analysis. In order to simulate setups of a specific type, do the following:
  1. In the Project Manager window, right-click on **Optimetrics**.
  2. Click **Analyze>All{*TYPE*}** from the shortcut menu where *TYPE* is the specific analysis type of interest, Parametric, Optimization, Sensitivity, or Statistical.

Viewing Analysis Results for Optimetrics Solutions

To view data specific to an Optimetrics solution, in general, do the following:

- In the project tree, right-click the Optimetrics setup for which you want to view the results, and then click **View Analysis Result** on the shortcut menu.

  The **Post Analysis Display** dialog box appears.
- Select from available setups by using the dropdown selection menu.



- Select the **Result** tab to view results in plot or table form. When you view results in Table form you can resort the results based on each column. For example, click the Variation column head to invert the sort from lowest to highest setup number. Click the variable name column to resort the results by step value. Click the Cost column head to sort the results from lowest cost to highest cost. Clicking a column again inverts the current sort.

- Click the **Options** button to open a dialog that permits you to specify the Maximum number of significant digits to display when showing the analysis result. The default is 4.
- Select the **Profile** tab to view start, stop, and elapsed times for each variable, and the analysis machine for each variation. You can click the column heads to sort the table by variation number,variable value, start, stop, or elapsed time, or (if you have run a distributed analysis) machine.

See the help topics in this section for more details about viewing optimization analysis results.

Related Topics

Viewing Solution Data for an Optimetrics Design Variation
Viewing an Optimetrics Solution's Profile Data
Viewing Results for Parametric Solution Quantities
Viewing Cost Results for an Optimization Analysis
Viewing Output Parameter Results for Sensitivity Analysis
Viewing Distribution Results for Statistical Analysis

Viewing Solution Data for an Optimetrics Design Variation

To view the convergence information, computing resources used, or matrices computed for any design variation solved during an optimization analysis, you must first select the design variation in the **Set Design Variation** dialog box. This dialog box is accessible from the **Solutions** Data window and via the **Results>Apply Solved Variation** command in the **Maxwell3D, Maxwell2D, or RMxprt menu**.

1. Click **Maxwell3D, Maxwell2D, or RMxprt and then select Results>Solution Data**.
   The **Solutions** dialog box appears.
2. Click the browsing dots beside the **Design Variation** box.
   The **Set Design Variation** dialog box appears.
3. Clear the **Use nominal design** option.
4. Click the design variation for which you want to view the solution data, and then click **OK**.
   The solution data is displayed in the table.

Related Topics

Viewing an Optimetrics Solution's Profile Data

Viewing an Optimetrics Solution's Profile Data

At any time during or after the Optimetrics solution process, you can see an overview of the computing resources or profile data that was used by Maxwell as it solved each design variation. The profile data indicates the how long each design variation took to solve.

1. In the project tree, right-click the Optimetrics solution setup of interest, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. Click the **Profile** tab.
3. Select the Optimetrics setup with the results you want to view from the pull-down list at the top of the dialog box.
4. Optionally, to examine more detailed profile data for a specific design variation, do the following:
   a. Click a design variation in the table.
   b. Click **Solver Profile**.

The **Solutions** dialog box appears with the profile data for the selected design variation.

The profile line for the matrix solver is in the following format:

```
Solver 123
```

where:

- *1* is the precision type: M (mixed) or D (double)
- *2* is the matrix data type: R (real) or C (complex)
- *3* is the symmetry type: S (symmetric), A (asymmetric), H (hermitian)

Related Topics

Viewing a Solution's Profile

Viewing Solution Data for an Optimetrics Design Variation

Viewing Results for Parametric Solution Quantities

1. In the project tree, right-click the parametric setup for which you want to view the results calculated for the solution quantities, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. Select the parametric setup with the results you want to view from the pull-down list at the top of the dialog box.
3. If it is not already selected, select **Table** as the view type.

   The results for the selected solution quantities are listed in table format for each solved design variation. The variation column in the table lists the entries in order. Clicking the Vision header inverts the order. Clicking other headers sorts the entries by value, and clicking again inverts the order.
4. Optionally, select **Show complete output name**.

   The complete name of the solution for which the results are being displayed will be listed in the column headings.

5. Optionally, click a design variation in the table, and then click **Apply** (at the far right side of the dialog box).

   The design displayed in the **3D Modeler** window is changed to represent the selected design variation.

Related Topics

Plotting Solution Quantity Results vs. a Swept Variable

Plotting Solution Quantity Results vs. a Swept Variable

To plot solution quantity results versus a swept variable's values on a rectangular (x - y) plot:

1. In the project tree, right-click the parametric setup for which you want to view the results, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. If it is not already selected, select **Plot** as the view type.
3. Select the variable with the swept values you want to plot on the x-axis from the **X** pull-down list.
4. Only one sweep variable at a time can be plotted against solution quantity results. Any other variables that were swept during the parametric analysis remain constant.

   Optionally, to modify the constant values of other swept variables, do the following:

   a. Click **Set Other Sweep Variables Value**.
      The **Setup Plot** dialog box appears. All of the other solved variable values are listed.
   b. Click the row with the variable value you want to use as the constant value in the plot, and then click **OK**.
5. Select the solution quantity results you want to plot on the y-axis from the **Y** pull-down list.

   The xy plot appears in the view window.
6. Right-click in the plot area to get the shortcut menu where you can set modify the plots display properties, print, copy to the clipboard, or export the data to a file.

Viewing Cost Results for an Optimization Analysis

To view cost values versus completed iterations in data table format:

1. In the project tree, right-click the optimization setup for which you want to view the cost results, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. Under the **Result** tab, select **Table** as the view type, if it is not already selected.

   The cost value at each solved design variation is listed in table format.
3. Optionally, click a design variation in the table, and then click **Apply**.

Maxwell now points to the selected design variation as the nominal solution and as a result, the design displayed in the **Modeler** window is changed to represent the selected design variation.

Click **Revert** to return the design in the view window to the original value.

Related Topics

Plotting Cost Data for an Optimization Analysis

Viewing Solution Data for an Optimetrics Design Variation

Plotting Cost Results for an Optimization Analysis

To view cost values versus completed iterations in rectangular (x-y) plot format:

1. In the project tree, right-click the optimization setup for which you want to view the cost results, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. Under the **Result** tab, select **Plot** as the view type.

   A plot of the cost value at each iteration appears.

Viewing Output Parameter Results for a Sensitivity Analysis

To view actual output parameter values versus design point in data table format:

1. In the project tree, right-click the sensitivity setup for which you want to view the parameter results, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. Under the **Result** tab, select **Table** as the view type, if it is not already selected.

   The following values are listed in table format:
   - The regression value of the output parameter at the design point is listed in the **Func. Value** column.
   - The first derivative of the regression is listed in the **1st D** column.
   - The second derivative of the regression is listed in the **2nd D** column.
3. Click **Apply**.

   Maxwell now points to the selected design variation as the nominal solution and as a result, the design displayed in the **Modeler** window is changed to represent the selected design variation.

   Click **Revert** to return the design in the view window to the original value.

Related Topics

Plotting Output Parameter Results for a Sensitivity Analysis

Viewing Solution Data for an Optimetrics Design Variation

Plotting Output Parameter Results for a Sensitivity Analysis

To plot output parameter results versus sensitivity variable values on a rectangular (xy) plot:

1. In the project tree, right-click the sensitivity setup for which you want to view the output parameter results, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. Under the **Result** tab, select **Plot** as the view type.
3. Select the sensitivity variable with the sweep values you want to plot on the x-axis from the **X** pull-down list.
4. Select the output parameter results you want to plot on the y-axis from the **Y** pull-down list.

   The xy plot appears in the **Post Analysis Display** dialog box.

   The plot displays actual output parameter results for each solved design variation. It also displays a parabola that best fits these results. The parabola is a more accurate representation of sensitivity around the design point than any individual solved design variation.

Viewing Distribution Results for a Statistical Analysis

1. In the project tree, right-click the statistical setup for which you want to view the distribution results calculated for the solution quantities, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. Select the statistical setup with the results you want to view from the pull-down list at the top of the dialog box.
3. To view the results in tabular form, select **Table** as the view type.

   The distribution results for the selected solution quantities are listed in table format for each solved design variation.
4. Optionally, click a design variation in the table, and then click **Apply** (at the far right side of the dialog box).

   The design displayed in the **3D Modeler** window is changed to represent the selected design variation.
5. To view the results in graphic format, select **Plot** as the view type.
6. Type the number of bins you want to plot on the x-axis.
7. Select the solution quantity for which you want to plot distribution results on the y-axis from the **Y** pull-down list.

   A histogram plot appears in the **Post Analysis Display** dialog box. It displays the distribution of the selected solution quantity.
8. Optionally, click a design variation in the table, and then click **Apply** (at the far right side of the dialog box).

Maxwell now points to the selected design variation as the nominal solution and as a result, the design displayed in the **Modeler** window is changed to represent the selected design variation.

Click **Revert** to return the design in the view window to the original value.

Related Topics

Plotting Distribution Results for a Statistical Analysis

Viewing Solution Data for an Optimetrics Design Variation

Plotting Distribution Results for a Statistical Analysis

1. In the project tree, right-click the statistical setup for which you want to view the distribution results calculated for the solution quantities, and then click **View Analysis Result** on the shortcut menu.

   The **Post Analysis Display** dialog box appears.
2. Select the statistical setup with the results you want to view from the pull-down list at the top of the dialog box.
3. If it is not already selected, select **Plot** as the view type.
4. Type the number of bins you want to plot on the x-axis.
5. Select the solution quantity for which you want to plot distribution results on the y-axis from the **Y** pull-down list.

   A histogram plot appears in the **Post Analysis Display** dialog box. It displays the distribution of the selected solution quantity.

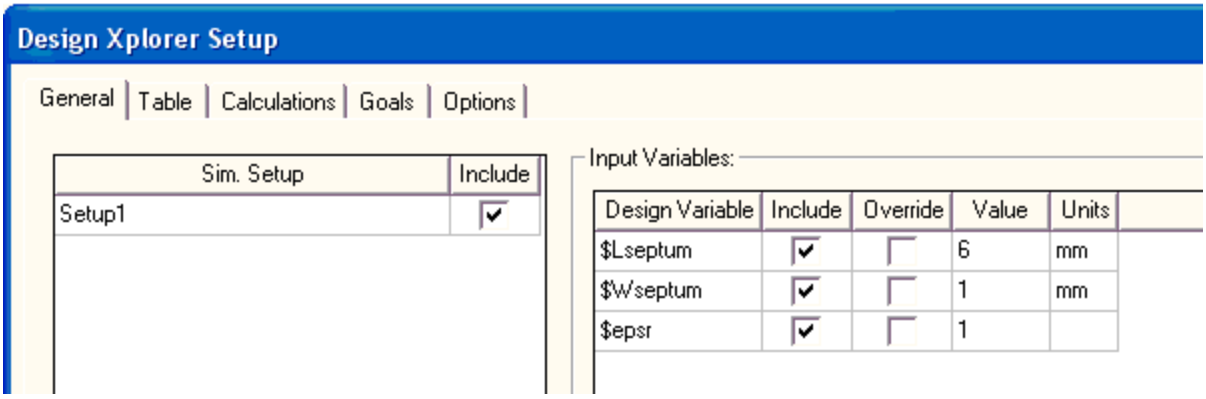
Link to Design Xplorer

You can export a .xml file containing information on a Maxwell setup, optimization variables, and output variables that enables Ansys Design Xplorer to manage Maxwell simulations, for example, for design of experiments and optimization. Design Xplorer will launch Maxwell simulations of design variations and evaluate the Maxwell outputs.

To do so:

1. Click **Maxwell>Optimetrics Analysis>Add Design Xplorer Setup...** or right-click on **Optimetrics** in the **Project** window, and select **Add Design Xplorer Setup...** from the short-cut menu.

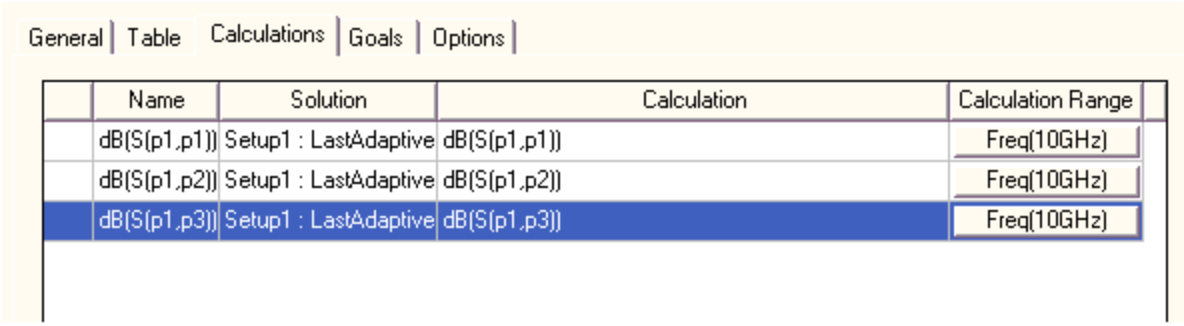
   This opens the **Design Xplorer** dialog with the **General** tab selected. it lists the setups available in the current project, and the input variables it contains.



2. Check Include for the simulation setups you want to use.
3. Check the Design variables to use. You can also chose to Override the value of a design variable. You can edit the Value and Units fields. Unchecking Override returns the values to their original state.
4. To setup any output calculations, click the **Calculation** tab and click the **Setup Calculations** button.

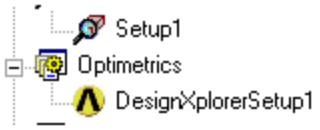
   This opens the **Add/Edit Calculation** dialog. Here you can define the simulation results of interest. The dialog box contains distinct panes and tabs to set the **Context**, the **Calculation Expression**, and the **Calculation Range**. See Setup Calculations for Optimetrics for details.

   Use the **Add Calculation** button to add expressions to the Calculations table of the **Design Xplorer Setup** dialog box, **Calculations** tab.

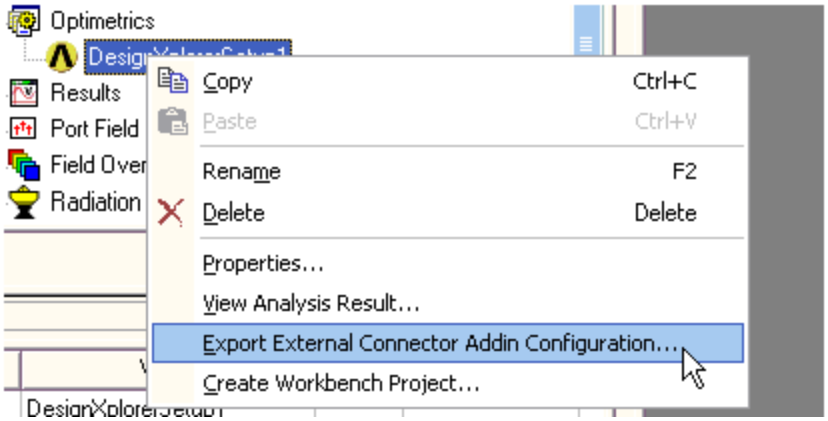


5. When you have added the calculations of interest, click **OK** to save the setup.

   An icon for the Design Xplorer setup appears under Optimization in the Project tree.



6. To create a .xml file with the setup information for Design Xplorer, first Save your project.
7. Then right-click on the setup and select **Export External Connector Addin Configuration.**



This displays a browser dialog that you can use to navigate your file system, and name and saves the .xml file. The .xml file contains information regarding the Maxwell path along with the setup, variables, and simulation results that you specified.

8. If you have an Ansys Workbench installation you can perform additional steps. Ensure that you have provided a path to the Workbench installation in the **Tools>General Options** dialog on the Miscellaneous panel.
9. Then click **Create Workbench Project**.

   This lets you name a Workbench project containing the information in the setup. The Ansys Workbench will be launched with the connection to the Maxwell project established. To this connection, you can add a Design Xplorer Setup. See the documentation of Ansys Workbench for details on Design Xplorer.

26 - Getting Started with RMxprt

RMxprt™ (Rotating Machines expert) is a template-based electrical machine design tool that provides fast, analytical calculations of machine performance and 2-D and 3-D geometry creation for detailed finite element calculations in Ansys® Maxwell®.

Using RMxprt, you can simulate and analyze the following types of machines:

- Three-phase and single-phase induction motors.
- Three-phase synchronous machines.
- Brushless permanent-magnet DC motors.
- Adjust-speed synchronous motors and generators.
- Permanent-magnet DC motors.
- Switched reluctance motors.
- Line-start permanent-magnet synchronous motors.
- Universal motors.
- General DC machines.
- Claw-pole alternators.
- Three-Phase Non-Salient Synchronous Machine
- Generic Rotating Machine

When you start to design a new model in RMxprt, you first select one of the above motor or generator types. You then enter the parameters associated with that machine type in each RMxprt Properties window. The properties windows are accessed by clicking each of the machine elements (for example, stator, rotor, shaft) under **Machine** in the project tree. General options are available directly at the **Machine** level of the project tree. Solution and output options (such as the rated output power) are set when you add a solution setup (by right-clicking **Analysis** in the project tree).

Related Topics

The RMxprt Desktop

RMxprt Commands

Setting Up A Machine Model

Creating a New RMxprt Project

Specifying RMxprt Machine Data

Getting Started with RMxprt

RMxprt™ (Rotating Machines expert) is a template-based electrical machine design tool that provides fast, analytical calculations of machine performance and 2-D and 3-D geometry creation for detailed finite element calculations in Ansys® Maxwell®.

Using RMxprt, you can simulate and analyze the following types of machines:

- Three-phase and single-phase induction motors.
- Three-phase synchronous machines.
- Brushless permanent-magnet DC motors.
- Adjust-speed synchronous motors and generators.
- Permanent-magnet DC motors.
- Switched reluctance motors.
- Line-start permanent-magnet synchronous motors.
- Universal motors.
- General DC machines.
- Claw-pole alternators.
- Three-Phase Non-Salient Synchronous Machine
- Generic Rotating Machine

When you start to design a new model in RMxprt, you first select one of the above motor or generator types. You then enter the parameters associated with that machine type in each RMxprt Properties window. The properties windows are accessed by clicking each of the machine elements (for example, stator, rotor, shaft) under **Machine** in the project tree. General options are available directly at the **Machine** level of the project tree. Solution and output options (such as the rated output power) are set when you add a solution setup (by right-clicking **Analysis** in the project tree).

Related Topics

The RMxprt Desktop

RMxprt Commands

Setting Up A Machine Model

Creating a New RMxprt Project

Specifying RMxprt Machine Data

Creating a Project and Inserting a New RMxprt Design

To create a new project:

1. Click **File>New**.

   A new project is listed in the project tree. It is named Project*n* by default, where *n* is the order in which the project was added to the current session.

Project definitions, such as material assignments, are stored under the project name in the project tree.

2. Click **Project>Insert RMxprt Design** or select **RMxprt** from the **Maxwell** pull-down menu on the **Desktop** ribbon.

   The **Select Machine Type** window appears.

3. Select the machine type you want, and click **OK**.

Specify the name of the project when you save it using the **File>Save** or **File>Save As** commands.

Opening Existing RMxprt Projects and Saving as New

You may also create new projects from existing ones, by saving them under new file names.

To create a new project from an existing one:

1. If you are already in the existing project, click **File>Save As**. The **Save As** window appears. (Otherwise, open the existing project you want to copy first.)
2. Enter a new name for the new project, and click **Save**.

The new project is now saved, with the same information as the existing project.

Opening RMxprt Projects

Open a previously saved project using the **File>Open** command.

1. Click **File**>**Open**.

   The **Open** dialog box appears.

2. Use the file browser to find the RMxprt version 6 project file.

   By default, files that can be opened or translated by RMxprt are displayed.

3. Select the file you want to open.
4. Click **OK**.

   The project information appears in the project tree.

Opening Recent RMxprt Projects

To open a project you recently saved:

- Click the name of the project file at the bottom of the **File** menu.

Saving RMxprt Projects

Use the **File>Save As** command to do the following:

- Save a new project.
- Save the active project with a different name or in a different location.

- Save the active project in another file format for use in another program.

Use the **File>Save** command to save the active project.

Related Topics

Saving a New Project
Saving the Active Project
Saving a Copy of a Project

Saving a New RMxprt Project

1. Click **File>Save As**.
   The **Save As** dialog box appears.
2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** box.
   By default, all files will have the **.aedt** extension.
1. Click **Save**.
   Ansys Electronics Desktop saves the project to the location you specified.

Related Topics

Saving the Active Project
Saving a Copy of a Project

Saving the Active RMxprt Project

- Click **File>Save**.
  RMxprt saves the project over the existing one.

Warning	Be sure to save machine models periodically. Saving frequently helps prevent the loss of your work if a problem occurs. Although RMxprt has an "auto-save" feature, it may not automatically save frequently enough for your needs.

Related Topics

Saving a New Project
Saving a Copy of a Project

Saving a Copy of an RMxprt Project

To save an existing, active project with a new name, a different file extension, or to a new location:

1. Click **File>Save As**.
2. Use the file browser to find the directory where you want to save the file.
3. Type the name of the file in the **File name** box.
4. Click **Save**.

   Ansys Electronics Desktop saves the project with the new name or file extension to the location you specified.

Related Topics

Saving a New Project
Saving the Active Project

Saving RMxprt Project Data Automatically

RMxprt stores recent actions you performed on the active project in an Ansys Electronics Desktop auto-save file in case a sudden workstation crash or other unexpected problem occurs. The auto-save file is stored in the same directory as the project file and is named **Project*n*.aedt.auto** by default, where *n* is the order in which the project was added to the current session. Ansys Electronics Desktop automatically saves all data for the project to the auto-save file, except solution data. By default, Ansys Electronics Desktop automatically saves project data after every ten edits. An "edit" is any action you perform that changes data in the project or the design, including actions associated with project management, model creation, and solution analysis.

With auto-save activated, after a problem occurs, you can choose to re-open the original project file (**Project*n*.aedt**) in an effort to recover the solution data or to open the auto-save file.

To modify the auto-save settings:

1. Click **Tools>Options>General Options**.

   The **Options** dialog box appears.
2. In the **General>Desktop Configuration** panel, verify that **Do Autosave** is selected.

   This option is selected by default.
3. In the **Autosave interval** box, enter the number of *edits* that you want to occur between automatic saves. By default, this option is set at 10.

Note	Auto-save *always* increments forward; therefore, even when you undo a command, Ansys Electronics Desktop counts it as an edit.
4. Click **OK** to apply the specified auto-save settings.

   Once the specified number of edits is carried out, a "model-only" save occurs. This means that Ansys Electronics Desktop does not save solutions data or clear any undo/redo history.

   When Ansys Electronics Desktop auto-saves, an ".auto" extension is appended to the original project file name. For example, **Project1.aedt** will automatically be saved as **Projectn.aedt.auto**.

Warning	When you close or rename a project, Ansys Electronics Desktop deletes the auto-

	save file. Ansys Electronics Desktop assumes that you have saved any desired changes at this point.

Related Topics

Recovering Project Data in an Auto-Save File

Recovering RMxprt Project Data in an Auto-Save File

Following a sudden workstation crash or other unexpected problem, you can recover the project data in its auto-save file.

Warning	When you recover a project's auto-save file you *cannot* recover any solutions data; recovering an auto-save file means you will lose any solutions data that existed in the original project file.

To recover project data in an auto-save file, if Ansys Electronics Desktop containing an RMxprt design has unexpectedly crashed:

1. Launch Ansys Electronics Desktop from your desktop.
2. Click **File>Open**,.
3. Select the original Project*n*.aedt project file for which you want to recover its Project*n*.aedt.auto auto-save file.

   The **Crash Recovery** window appears, giving you the option to open the original project file or the auto-save file.
4. Select **Open project using autosave file** to recover project data in the auto-save file**, and then click OK.** Ansys Electronics Desktop replaces the original project file with the data in the auto-save file.

   Ansys Electronics Desktop immediately overwrites the original project file data with the auto-save file data, removing the results directory (solutions data) from the original project file as it overwrites to the auto-save file.

Warning	If you choose to recover the auto-save file, you cannot recover the original project file that has been overwritten; recovering data in an auto-save file is *not* reversible.

Related Topics

Saving Project Data Automatically

RMxprt Files

When you create any design in the Ansys Electronics Desktop, including an RMxprt design, it is given an **.aedt** file extension and stored in the directory you specify. Any files related to that project are also stored in that directory.

Some common file and folder types are listed below:

.aedt	Maxwell or RMxprt project.
*project_name.***aedtresults**	Folder containing results data for a project.
*design_name.***results**	Folder containing results data for a design. This folder is stored in the *project_name.***aedtresults folder.**
*design_name.***asol**	Results data for a design. This file's contents may be empty if a solution is unavailable. This file is stored in the *project_name.***aedtresults** folder.

Saving Project Notes in RMxprt

You can save notes about a project, such as its creation date and a description of the device being modeled. This is useful for keeping a running log on the project.

To add notes to a project:

1. Click **RMxprt**>**Edit Notes**.

   The **Design Notes** dialog box appears.
2. Click in the window and type your notes.
3. Click **OK** to save the notes with the current project.

To edit existing project notes:

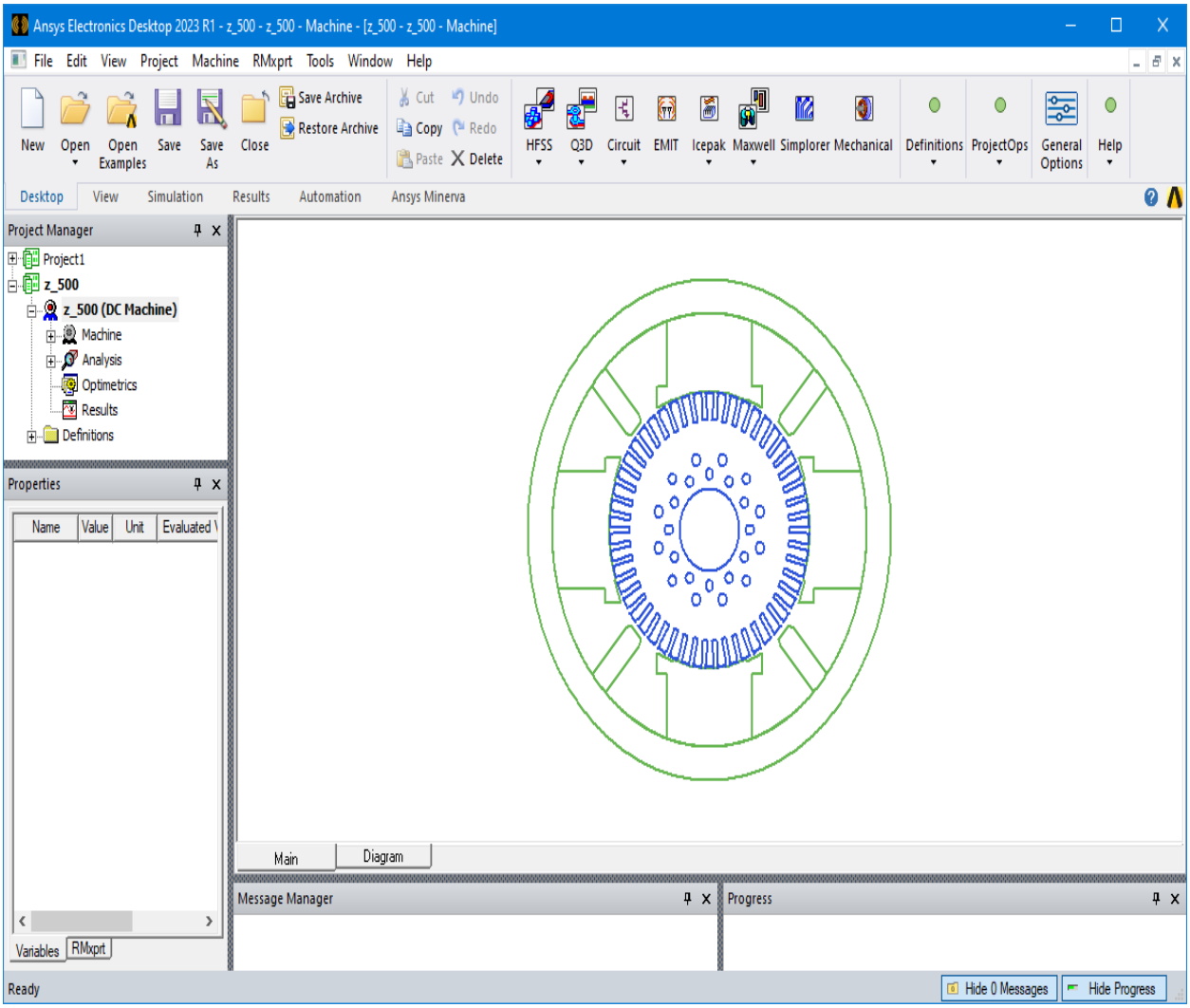
1. Double-click the **Notes** icon in the project tree.
   The **Design Notes** window appears, where you can edit the project's notes.
2. Click **OK** to save any changes, or click **Cancel** to exit without saving edits.

The RMxprt Desktop

RMxprt is integrated within the Maxwell desktop. Consistent with the Maxwell desktop, the RMxprt interface consists of several desktop components: a title bar, a menu bar, ribbons, a status bar, a project manager window, a properties window, a message manager window, a progress window, and a machine editor window. If user-defined rotor or stator slots are used in the design, a slot editor window also displays when a rotor or stator slot is selected in the project tree. The project manager window, the properties window, the message manager window and the progress window are dockable and resizeable.

You can open multiple machine editor windows to display different parts at the same time. One can remain fixed on the winding, one on the diagram, and one on the main desktop window. To open a new window, click **Window>New Window**.

To move back and forth between windows, select the window you want to view from the **Windows**.



RMxprt Title Bar

The title bar is located at the top of the application window. It displays the information of the active design. If a machine editor window is maximized, its title is appended in the title bar within square brackets. The information of the active design includes the desktop name, the project name, the design name and the design type. For an RMxprt design, the design type is **Machine**.

Working with the RMxprt Menu Bar

The menu bar enables you to perform all Maxwell, ePhysics, and/or RMxprt tasks, depending on the software you purchased. Such tasks include managing project files, customizing the desktop, drawing objects, and setting and modifying all project parameters.

RMxprt contains the following menus, which appear at the top of the desktop:

File menu	Use the **File** menu commands to manage RMxprt project files and printing options.
Edit menu	Use the **Edit** menu commands to modify properties in the active design, manage designs in one or more projects, delete projects, and undo and redo actions.
View menu	Use the **View** menu commands to display or hide desktop components, and change the machine editor window view.
Project menu	Use the **Project** menu commands to add a Maxwell 3D, Maxwell2D, or RMxprt design to the active project, analyze all designs of the active project, and define project variables and datasets.
Machine menu	Use the **Machine** menu to work with the machine data, such as edit winding layout, edit wire size, and set dimension unit for the active editor window.
RMxprt menu	Use the **RMxprt** menu commands to validate design input data, analyze designs, set up parameters, add analysis setups, set up Optimetrics, post process solutions, and other design tasks.
Tools menu	Use the **Tools** menu to modify the active project's material library, arrange the material libraries, run and record scripts, update project definitions from libraries, display options, and modify many of the software's default settings.
Window menu	Use the **Window** menu commands to rearrange the application windows.
Help menu	Use the **Help** menu commands to access the help system and view the current software version information.

Working with the RMxprt Shortcut Menus

A variety of shortcut menus — menus that appear when you right-click a selection — are available in the ribbons area of the desktop, in the **Machine Editor** window, in the **Project Manager** window, in the **Properties** window, and in the **Message Manager** window.

In Machine Editor window	Use the shortcut menu in the **Machine Editor** window to edit winding layout, display or hidden coil connection, change the view, and copy to Clipboard.
In the Slot Editor window	Use the shortcut menu in the **Slot Editor** window to insert, append, modify, and remove slot segments.
In the Project Manager window	Use the shortcut menus in the **Project Manager** window (or the project tree) to manage project files and design properties; these commands duplicate menu commands at the top of the screen.
In Properties window	Use the shortcut menus in the **Properties** window to edit (cut, copy, paste or delete) property values.

In Message Manager window	Use the shortcut menus in the **Message Manager** window to clear, copy message, or see message details.

Note	Most of the commands on the shortcut menus are also available on the menu bar.

Undoing RMxprt Commands

Use the **Undo** command on the **Edit** menu to cancel, or undo, the last action you performed on the active project or design.

1. In the **Project Manager** window, do one of the following:
   - To undo the last action you performed on the *active project*, such as inserting a design, click the project icon.
   - To undo the last action you performed on the *active design*, click the design icon.

Note	**You cannot undo an analysis that you have performed on a model, that is, the RMxprt>Analyze command.**

2. Click **Edit>Undo**.

   Your last action is now undone.

Note	When you save a project, RMxprt always clears the entire undo/redo history for the project and its designs.

Related Topics

Redoing Commands

Redoing RMxprt Commands

Use the **Redo** command on the **Edit** menu to reapply, or redo, the last action that was canceled, or undone. **You can redo a canceled action related to project management, model creation, and post-processing.**

1. In the **Project Manager** window, do one of the following:
   - To redo the last action you canceled on the *active project*, such as inserting a design or adding project variables, click the project icon.
   - To redo the last action you canceled on the *active design*, such as drawing an object or deleting a field overlay plot, click the design icon.
2. Click **Edit>Redo**.

   Your last canceled action is now reapplied.

Note	When you save a project, RMxprt always clears the entire undo/redo history for the project and its designs.

Related Topics

Undoing Commands

Working with the RMxprt Status Bar

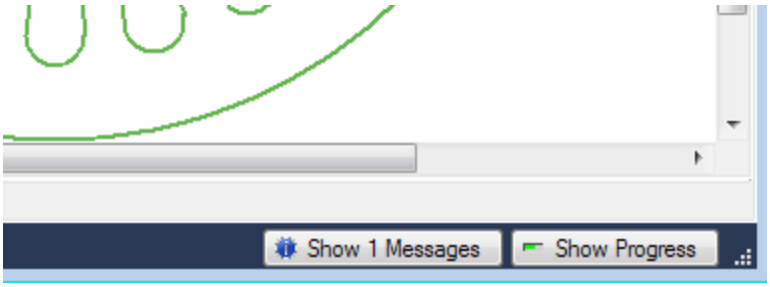
The status bar is located at the bottom of the application window. It displays information about the command currently being performed.

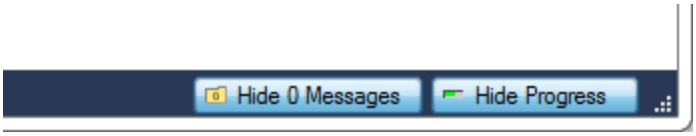
To display or hide the status bar:

- Click **View>StatusBar**.

  A check box appears next to this command if the status bar is visible.

  When visible, the status bar contains buttons to show or hide the message window and progress window.





Working with the RMxprt Machine Editor Windows

You can open multiple machine editor windows in RMxprt. One can remain fixed on the **Winding Editor**, one on the **Diagram** tab, and one on the **Main** tab. To open a new window, click **Window>New Window**. To move back and forth between windows, select the **Windows** menu, and select the window you want to view.

You can cascade all **Machine Editor** windows, tile them horizontally or vertically. You can maximize, minimize or close a **Machine Editor** window by clicking the relevant button on the right-top corner of the window. If no **Machine Editor** window is displayed, you can use **RMxprt>Machine Editor** to bring one window up. When only one **Machine Editor** window is maximized, the window title is displayed within square brackets in the **Title Bar** of the main application window.

As you enter appropriate property values, the **Machine Editor** window dynamically updates the rotor, stator, slots, and windings in the **Main**, **Diagram** and **Winding Editor** tabs. As you provide winding information, the **Winding Editor** tab displays a table of values.

Related Topics

Setting the Window View
Printing in RMxprt

Setting the Window View

To fit the entire diagram in the window:

- Click **View>Fit All**.

To zoom into the diagram in the window:

- Click **View>Zoom In**.

To zoom out of the diagram in the window:

- Click **View>Zoom Out**.

Printing in RMxprt

The printing commands enable you to print the display in the active window.

To print the project:

1. Click **File**>**Print**.
   The **Print** dialog box appears.
2. You can change the print quality (a higher dpi produces a higher quality print but takes more time and printer memory), or you can send the output to a **.prn** file.
3. Do one of the following:
   - Click **OK** to print the project.
   - Click **Cancel** to dismiss the window without printing.
   - Click **Properties** to define printer settings.

Working with the RMxprt Project Manager

The **Project Manager** window displays the open project's structure, which is referred to as the project tree. The **Project Manager** window displays details about all projects open in the Maxwell Desktop, regardless of type.

To show or hide the **Project Manager** window, do one of the following:

- Click **View>Project Manager**.
  A check box appears next to this command if the **Project Manager** window is visible.

Related Topics

Working with the RMxprt Project Tree
Shortcut Menus in the Project Manager Window
"Working with the Project Manager" on page 2-17

Working with the RMxprt Project Tree

The project tree is located in the **Project Manager** window and contains details about all open projects. The top node listed in the project tree is the project name. It is named Project*n* by default, where *n* is the order in which the project was added to the current session of the Maxwell Desktop. Expand the project icon to view all designs and material definitions belonging to the project. For RMxprt projects, the project tree shows where you can select each portion of the machine to open the corresponding tab sheet in the P**roperties** window. The project tree lists options for the general motor characteristics, the stator, the rotor, and other options such as winding data or commutating data. The specific options depend on the machine type you have selected.

Related Topics

Viewing RMxprt Design Details
Automatically Expand the Project Tree

Setting the RMxprt Project Tree to Expand Automatically

You can set the project tree to automatically expand when an item is added to a project.

1. Click **Tools>Options>General Options**.

   The **Options** dialog box appears.

2. Click the **Project Options** tab.
3. Under **Additional Options**, select **Expand Project Tree on Insert**.
4. Click **OK**.

Viewing RMxprt Design Details

Once you insert an RMxprt design into a project, it is listed as the second-level node in the project tree. It is named RMxprtDesign*n* by default, where *n* is the order in which the design was added to the project. Expand the design icon in the project tree to view specific data about the model.

The RMxprtDesign*n* node contains the following project details:

Machine	Allows you to specify parameters for various aspects of the machine. A whole or part geometry will be drawn in the **Main** tab of the **Machine Editor** window

	(based on the values you enter).
Analysis	Displays the solution setups for an RMxprt design. A solution setup specifies how RMxprt computes the solution.
Optimetrics	Displays any Optimetrics setups added to an RMxprt design.
Results	Displays any post-processing reports that have been generated.

Note	To edit a project's design details: • In the project tree, double-click the design setup icon that you want to edit. A dialog box appears with that setup's parameters, which you can then edit.

Working with the RMxprt Properties Window

The **Properties** window displays the attributes, or properties, of an item selected in the project tree and enables you to edit an item's properties. The properties, and the ability to edit them in the **Properties** window vary depending on the type of item selected. The tabs available in the **Properties** window also vary depending the selection.

Single clicking on an item in the Machine section of the project tree displays a docked **Properties** window located under the project tree. A horizontal scroll bar lets you adjust the view of the properties if necessary. Changes to values in the docked properties window apply immediately to the selected object.

Double-clicking on an item in the Machine section of the project tree opens a floating **Properties** window. The floating window can be moved for convenience in viewing the RMxprt **Machine Editor** window. Some objects have tabs on the window to control the properties displayed. Changes to values in the floating window are not applied until you click the **OK** button.

Related Topics

Showing and Hiding the Properties Window

Setting the Properties Window to Open Automatically

Showing and Hiding the RMxprt Properties Window

To show or hide the **Properties** window on the desktop:

- Click **View>Properties**.

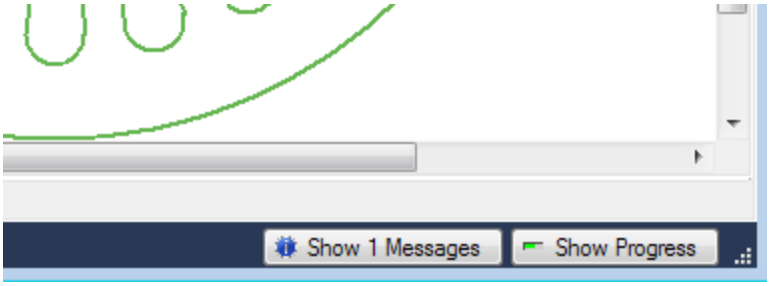
A check box appears next to this command if the **Properties** window is visible.

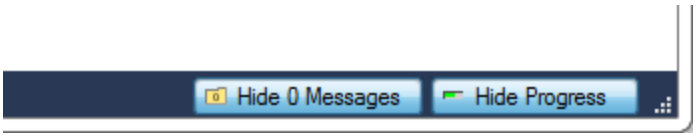
Working with the RMxprt Progress Window

The **Progress** window monitors a simulation while it is running.

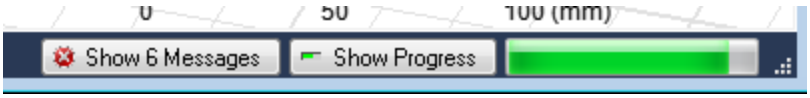
To display or hide the **Progress** window on the desktop, do one of the following:

- Click the **Show Progress** or **Hide Progress** buttons on the status bar:





When more than one progress bar is active, the top progress bar is represented on the status bar with a progress indicator.



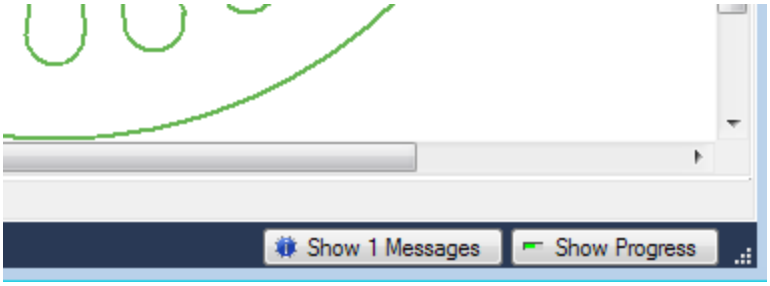
- Click **View>Progress Window**.
  A check box appears next to this command if the **Progress** window is visible.

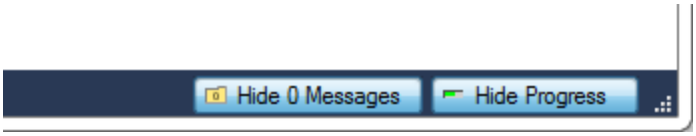
Working with the RMxprt Message Manager

The **Message Manager** displays messages associated with a project's development, such as error messages about the design's setup or informational messages about the progress of an analysis.

To display or hide the **Message Manager** window on the desktop, do one of the following:

- Click the **Show Messages** or **Hide Messages** buttons on the status bar:





- Click **View>Message Manager**.
  A check box appears next to this command if the **Message Manager** is visible.

Related Topics

Clearing Messages for the RMxprt Project
Clearing Messages for the RMxprt Model
Copying Messages in RMxprt

Clearing Messages for the RMxprt Project

You can clear all the messages for a particular project.

To clear messages:

1. Right-click the *project#* in the **Message Manager**.
   A pop-up appears.
2. Click **Clear messages for Project#**.

Clearing Messages for the RMxprt Model

You can clear all the messages for a particular model.

To clear messages:

1. Right-click the *RMxprtDesign#* in the **Message Manager**.
   A pop-up appears.
2. Click **Clear messages for RMxprtDesign#**.

Copying Messages in RMxprt

You can copy all the messages for a particular project.

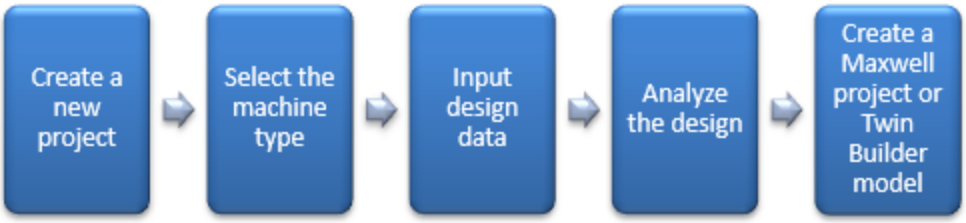
To copy messages:

1. Right-click in the **Message Manager**.
   A pop-up appears.
2. Click **Copy messages to clipboard**.

Quick Start for RMxprt

This section briefly introduces how to enter the environment of **RMxprt** and quickly master its main functions by providing a simple example.

The basic process flow chart is shown below.



RMxprt Example Part 1: Create a New Project

To open Ansys Electronics Desktop and create a new project:

1. Double-click the **Ansys Electronics Desktop** icon on your desktop to launch Ansys Electronics Desktop.

   You can also start Ansys Electronics Desktop by clicking **Start** > **All Programs** > **Ansys EM Suite** [version] >**Ansys Electronics Desktop** [version] from Windows.
2. Click **File>New** from the menu bar.

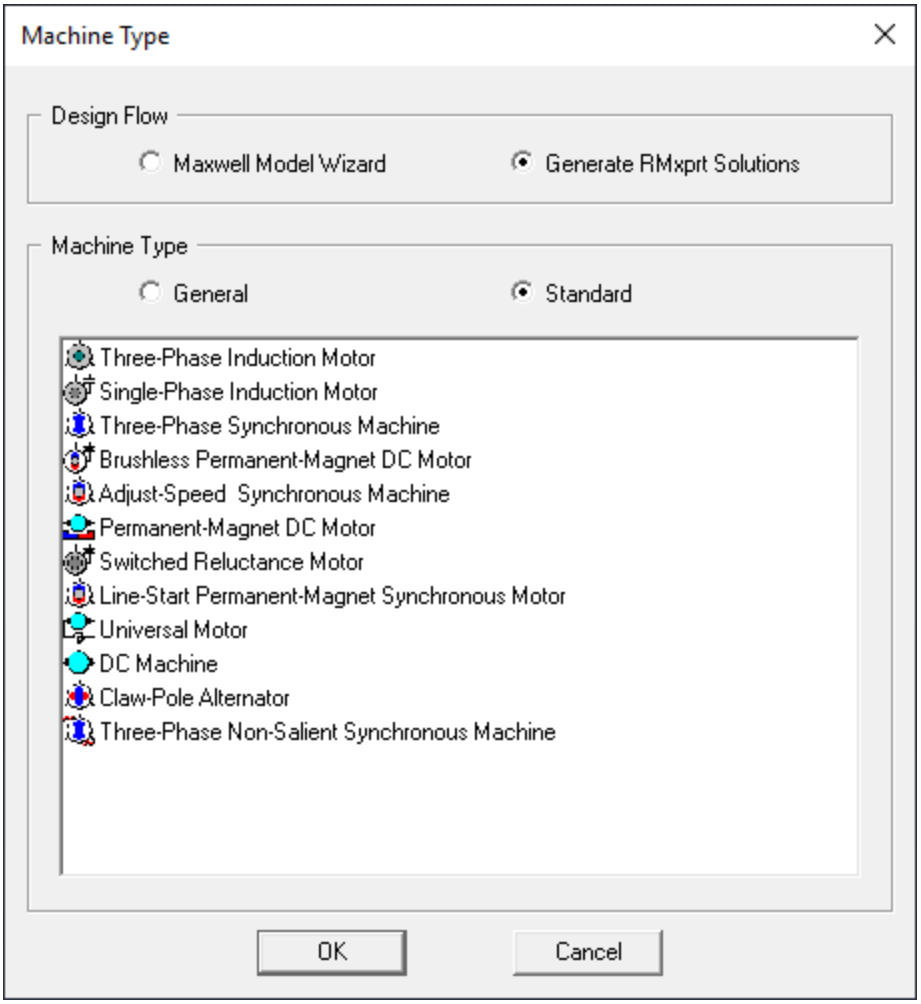
   This creates a new project folder in the project window with the default name of Project*n.*

RMxprt Example Part 2: Select a Machine

To select a machine to insert into the new project:

1. Click **Project>Insert RMxprt Design** or select the RMxprt icon on the Desktop ribbon.

   This displays the **Machine Type** window.



2. For the Design Flow, select the **Generate RMxprt Solutions** radio button; then select **Standard** for Machine Type.From the list of machine types, for this example, select **Brushless Permanent Magnet DC Motor** and click **OK**.

   This closes the window and inserts the Brushless Permanent Magnet DC Motor design in the project.

Continue to Part 3 of the example to Input Design Data.

RMxprt Example Part 3: Input Design Data

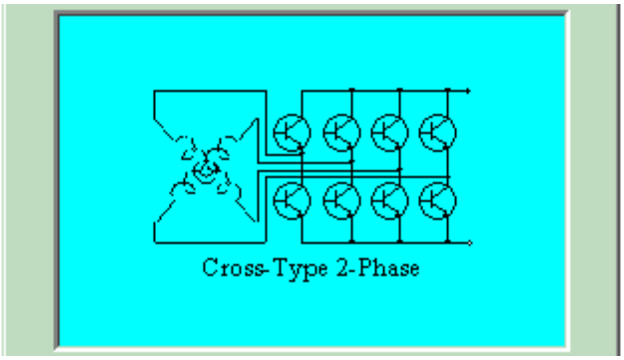
In this part of the example, you provide values for the design and for various parts.

1. Click the + symbol by the RMxprt:Design*n* icon in the project tree to view the design hierarchy.

   This displays the Machine Icon.

2. Double-click the icon to view the Machine Properties window.

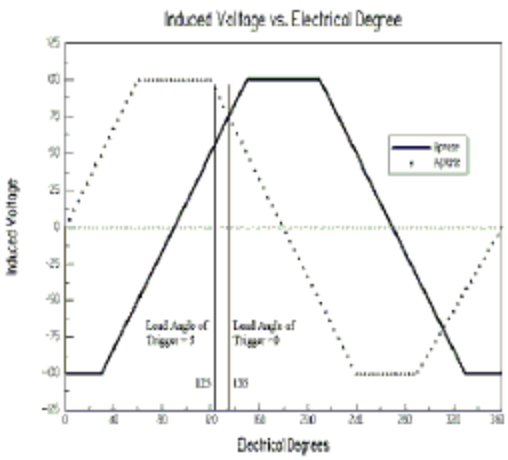
   Set the values as indicated below.

Machine Type	Brushless Permanent Magnet DC Motor
Number of Poles	Set this to 4
Rotor Position	Set to Inner Rotor
Frictional Loss	Set this to 11 Watts (Frictional and windage loss is typically within the range of 1%~3% of the rated output power, in this example, 2% is estimated.) This value is referred to the given **Reference Speed**. The frictional loss at the computed rated speed will be modified if the computed rated speed is different from the given rated speed.
Windage Loss	0
Reference Speed	Set this to 1500 rpm
Control Type	DC
Circuit Type	Set this to C2. Click the button to display the Select Circuit Type window.  Select the C2 button, and OK to close the window.

3. Click OK to close the Machine properties window.
4. Click the + symbol by the Machine icon to view the design hierarchy of the motor.
5. Double-click the Circuit icon to view the Circuit properties window.

   Set the values as indicated below.

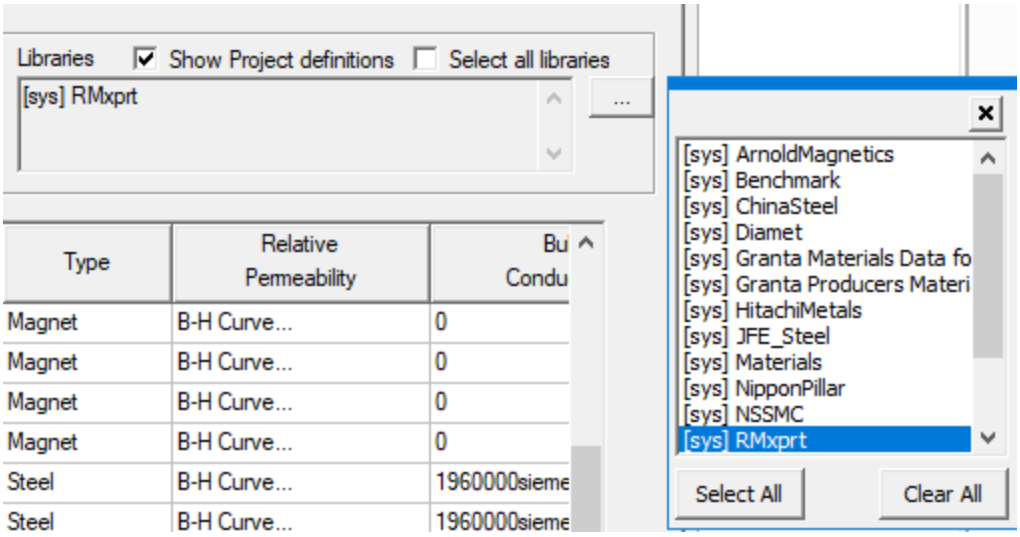
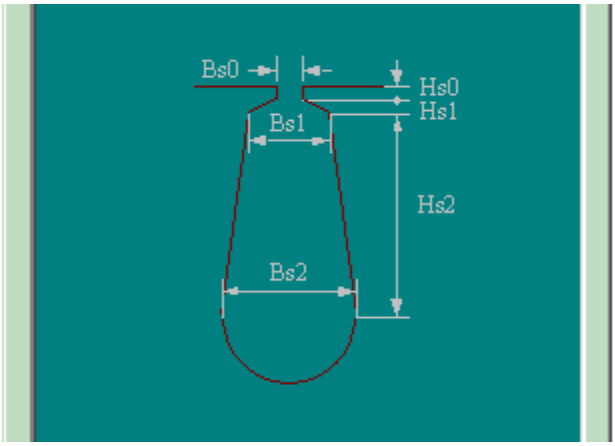
Lead Angle of Trigger	Set this to 0 degrees to obtain the maximum average emf for the following phase in the trigonal period.

	
Trigger Pulse Width	Set this to 90 degrees
Transistor Drop	Set this to 2 Volts
Diode Drop	Set this to 2 Volts

6. Click OK to close the circuit properties window.
7. Double-click the Stator icon to view the Stator properties window.

   Set the values as shown below.

Outer Diameter	Set this to 120 mm.
Inner Diameter	Set this to 75 mm.
Length	Set this to 65 mm for the length of the Stator iron core.
Stacking Factor	0.95
Steel Type	Click on the button to display the **Materials** window. Select **RMxprt** library in the **Libraries** box in the upper right corner of the **Materials** window: then select **M19-24G** from the list.

	 Note: If RMxprt is not listed in the libraries box in the upper right corner of the Materials window, quit the Materials window, click **Tools>Configure Libraries**, add **RMxprt** (under materials) and click the **Save as Default** check box. Then click OK.
Number of Slots	Set this to 24.
Slot Type	Select 2 as the Slot type. Click the button on the row cell to display the Select Slot Type window.  Click the 2 button and OK to close the window.
Skew Width	Set this to 1. (To skew one slot pitch.)

8. Click **OK** to close the **Stator Properties** window.

Take a moment to look at the Maxwell Design window. If you click the **Main** tab, you will see two concentric rings that represent the inner and outer diameters you specified. If you click the **Winding Editor** tab, you see a table of the coils, with columns for **Phase**, **Turns**, the **In Slots**, and the **Out Slots**. There is also a drawing showing the placement of the 24 slots of the type that you defined here.

9. Click the + symbol by the **Stator** icon to view the hierarchy under the stator.
10. Click the slot icon to view the **Slot Properties** window.

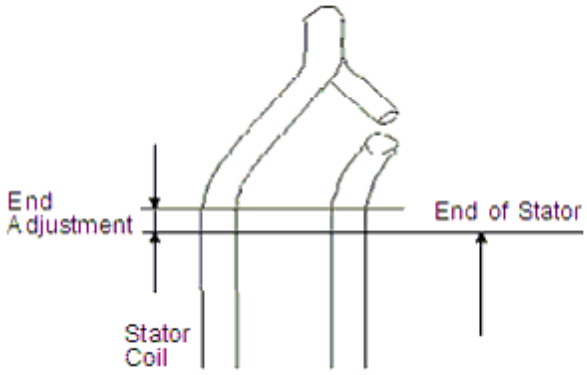
    Set the values as shown below. Some of the properties will not appear until you disable the **Auto Design** property in the first row.

Auto Design	Uncheck the box to disable auto design. Close the properties window and open it again. Then set the given values for the slot shapes.
Parallel Tooth	Uncheck this box. The Tooth Width property becomes invisible.
Tooth Width	Invisible when Parallel Tooth is unchecked.
Hs0	Set to 0.5 mm
Hs1	Set to 1.0 mm
Hs2	Set to 8.2 mm
Bs0	Set to 2.5 mm
Bs1	Set to 5.6 mm
Bs2	Set to 7.6 mm

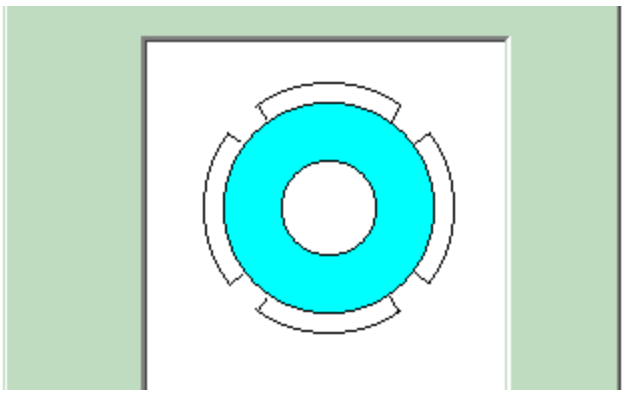
11. Click the stator **Winding** icon to view the winding **Properties** window.

    Set the values as shown below.

Winding tab	**Winding Layers**	Set this to 2.
	Winding Type	"Whole-Coiled."
	Parallel Branches	Enter 1 for the number of parallel-connected branches, i.e. the coils in all the slots per phase are in series-connected.
	Conductors per Slot	Set this to 60 for the number of conductors per slot, i.e. the number of turns per coil is equal to 30 for double-layer winding.
	Coil Pitch	Set this to 5. For this example, full pitch = 24 slots / 4 poles = 6. This example uses short coil pitch, *5*, i.e. a coil spans from slot 1 to slot 6.
	Number of Strands	Select *1* for the number of strands (or number of wires per conductor).
	Wire Wrap	Select *0*. This is the total thickness of double side wire insulation. The input value *0* means that **RMxprt** will automatically check into the wire gauge library for the wrap

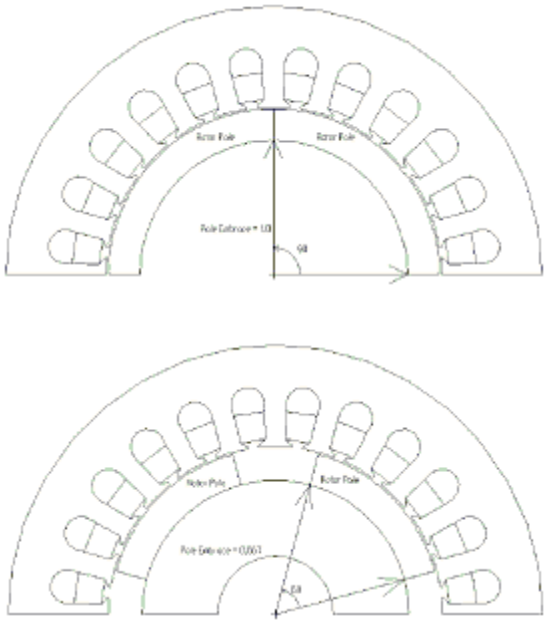
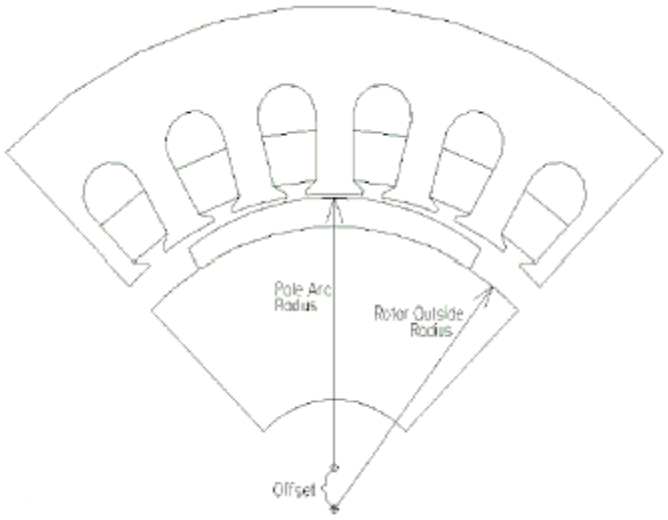
		thickness relevant to the wire gauge. Different manufacturers produce different **Wire Wrap Thickness** for electromagnetic wire. Typically, **Wire Wrap Thickness** for electromagnetic wire is 7~10% of **Wire Diameter**.
	Wire Size	Click on the Properties field to display the Wire Size window and select *AUTO* for automatic design of wire **Gauge**. Wire Size will be set to 0 in the Wire Size window. This example relies on **RMxprt** to automatically select the optimum diameter and the gauge code for electromagnetic wire.
End/Insulation tab	**Input Half-turn Length**	Uncheck this box.
	Half Turn Length	This item is not shown if **Input Half Turn Length** is unchecked.
	End Extension (End Adjustment)	Set this to 0 mm for the linear overhang of the end part of the coil out of the iron core as shown below. In this example, the coil turns immediately at the slot opening, therefore input 0**.** 
	Base Inner Radius	0
	Tip Inner Diameter	0
	End Clearance	0
	Slot Liner	Set this to 0.3 for the single side thickness of slot insulation.
	Wedge Thickness	0
	Layer Insulation	0
	Limited Fill Factor	0.75 (The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.)

13. Click **OK** to close the stator Winding Properties window.
14. Click **Machine>Winding>Connect All Coils** on the menu bar.
    The **Winding** tab in the main window shows all coils connected.
15. Double-click the Rotor Icon to view the **Rotor Properties** window.
    Set the values as shown below.

Outer Diameter	Set this to 74.0 mm. This is the Stator inner diameter - 2* AirGap.
Inner Diameter	Input 26 mm for the inner diameter of the rotor core. This is also the diameter to match the shaft.
Length	Input 65 mm for the length of the rotor core. In this example, the lengths of the iron cores of the stator and the rotor are the same.
Steel Type	Select M19-24G for the material of the silicon-steel sheet for the rotor. In this example, the laminations are punched together on the same sheet; therefore, the material of the silicon-steel sheet and the stacking factors are the same for the stator and the rotor.
Stacking Factor	Input 0.95.
Pole Type	Select 1. Click on the button on the Pole Type field to display the Select Pole Type window.  Click the 1 button and OK to close the window.

16. Click **OK** to close the Rotor Properties window.
17. Click the + symbol by the Rotor icon to open the project hierarchy under the rotor.
18. Double-click the Pole icon to view the Pole Properties window.
    Set the values as shown below.

Embrace	Input 0.7. **Embrace** of the rotor represents the ratio of the rotor central angle corresponding to the arc length along the rotor surface of an arched permanent-magnetic piece to the rotor central angle corresponding to a rotor pole. In a four

	pole machine with **Embrace**, 1, each arched permanent-magnetic piece covers 90 mechanical degrees along the rotor surface. Similarly, **Embrace** 0.667 means 60 mechanical degrees of the coverage of the magnet as shown in the figure. 
Offset	Input 0. The arched permanent-magnetic pieces to form the magnets of the rotor might not be concentric with the rotor as shown in the figure. In the electric machines with non-uniform air-gap, there exists an offset between the two centers. **RMxprt** terms it as **Pole Arc Offset**. This example uses uniform air-gap; therefore, the offset is set to *0*. 
Magnet	Select XG196/96. This permanent-magnetic steel possesses residual flux

Type	density 0.96 Tesla, coercive force 690 kA/m, maximum magnetic energy product 183 kJ/m^3, and relative recoil magnetic permeability 1.0.
Magnet Thickness	Input 3.5 mm for the thickness of the permanent-magnetic steel.

19. Click **OK** to close the Pole Properties window.

To continue to Part 4 of the example, go to Analyze the Design.

RMxprt Example Part 4: Analyze the Design.

Before analyzing a design project, a few options should be decided using the following procedures:

1. Click **Tools>Options>General Options** to open the **Options** dialog box, displaying a hierarchical list of options. Click the + next to **Machines** to show the following choice: **Wire Setting**.

   The System Libraries setting should be set to **American**.
2. Click **OK** to close the window.
3. Click **RMxprt>Analysis Setup>Add Solution Setup**.

   This displays the Solution Setup window. Add the following values.

Load Type	Const Power
Rated Output Power	550 W
Rated Voltage	220 V
Rated Speed	1500 rpm
Operating Temperature	75cel

4. Click **OK** to close the dialog to save the Setup.
5. Click **RMxprt>Validation Check** to ensure that all values have been set.

   If any items do not pass validation, use the diagnostic information in the Message Window to resolve any issues.
6. When the design has been validated, click **RMxprt>Analyze All**.

   The progress of the analysis is shown in the Progress window.

To continue to Part 5 of the example, go to Create Reports and View Output.

RMxprt Example Part 5: Create Reports and View Output

After you have run an analysis, you can view the solution data.

1. Click **RMxprt>Results>Solution Data**.

   This opens the **Solutions** window with the **Performance** tab selected, and the FEA Input Data displayed. The **Solutions** window contains tabs for the following:

   - **Performance** – the **Data** field in the **Performance** tab is a drop down menu from which you can select the following:
     - FEA Input Data
     - Full Load Operation
     - Material Consumption
     - No Load Operation
     - Permanent Magnet
     - Rotor Data
     - Stator Slot
     - Stator Winding
     - Steady State Parameters
   - **Design Sheet** – Shows the various design parameter values
   - **Curves** – Selecting the **Curves** tab lets you view predefined graphs.

2. On the **Performance**tab, select **Stator Winding** as the **Data** selection.

   Except for some data corresponding to the wire gauge, this data should be the same as the data input in the Stator Winding Properties window. Since automatic design function for the wire gauge is selected in the input, **RMxprt** calculates the following data:

Wire Diameter (mm):	*0.8118* for the diameter of the electromagnetic wire.
Wire Wrap (mm):	*0* for the insulation thickness of the electromagnetic wire. Because input wire wrap is *0*, RMxprt picks it up from the selected wire library (American wire), but it still *0* based on the wire wrap data in the library.
Stator Slot Fill Factor (%):	*61.6599*.

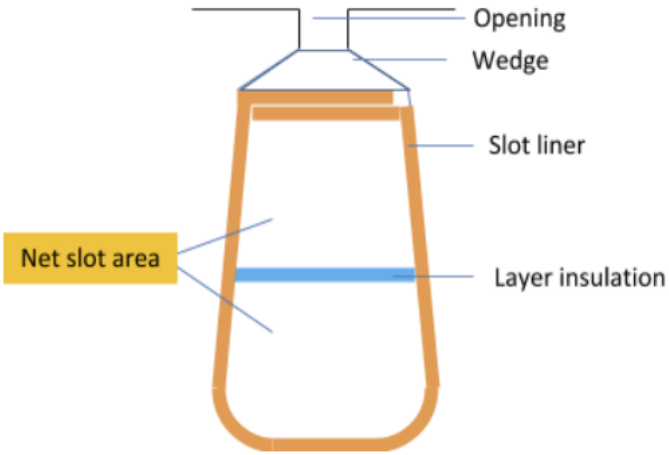
   The electromagnetic wire with **Wire Diameter** of *0.8118* is equivalent to AWG 20. **Stator Slot Fill Factor** represents the percentage of occupation of the slot area, i.e. the ratio of the total square sectional area of wires (including Wire Wrap thickness) in a slot to the total slot area less the slot insulation.

   The slot fill factor is defined as:

$$F_{SF} = \frac{n \cdot d^2}{S_{net}}$$

where:

- n is the total number of wires per slot (the number of conductors per slot times the number of wires per conductor),
- d^2 is the square area of a wire with d being the wire diameter including wire insulation, and
- S_{net} is the net slot area which is obtained from the gross slot area minus opening area, wedge area, liner area, and layer insulation area, as shown below:



a. Now that the **Wire Diameter** of the electromagnetic wire has been calculated by **RMxprt**, you can open the **Winding Properties** window and specify the value.
b. For **Wire Size,** open the Wire Size selection window, select *0.8118* for the electromagnetic wire diameter, which corresponds to *20* for the wire gauge.
c. In the slot **Wire Wrap** field, input *0.08* for the insulation thickness of the electromagnetic wire.
d. Click **OK** to close the properties window.
e. Click **RMxprt>Analyze All**.

   After the second analysis is completed, click **RMxprt>Results>Solution Data** to view the effect of **Wire Wrap Thickness** of the electromagnetic wire on **Stator Slot Fill Factor**.

Wire Diameter (mm):	*0.8118*.
Wire Wrap Thickness (mm):	*0.08*.
Stator Slot Fill Factor (%):	*75.6182*.

3. In the **Solutions** window, change the Data selection to **Rotor Data**.

   The Rotor data is displayed.

   Here most of the data is the same as input in the **Rotor Pole** properties window. The only difference is that the **Pole Arc Radius** replaces **Pole Arc Offset** and, in addition to **MechanicalPole Embrace** which is input based on the physical geometry, **Electrical Pole Embrace** is also given. **Electrical Pole Embrace** is calculated by the ratio of the average

magnetic flux density to the maximum magnetic flux density according to the magnetic flux density distribution along the air-gap.

4. In the **Solutions** window, change the Data selection to **Permanent Magnet**.

   This part displays the characteristic data of the permanent magnets as well as the **Demagnetized Flux Density**, the **Recoil Residual Flux Density**, and the **Recoil Coercive Force** of the recoil line based on the demagnetization flux density, which are used for finite element analysis when a linear PM characteristics must be specified.

5. In the **Solutions** window, change the Data selection to **Steady State Parameters**.

   This part displays the **Stator Winding Factor**, direct- and the quadratic-axis inductances, the **Armature Leakage Inductance**, the **Armature Phase Resistance** of the phase winding, the direct- and the quadratic-axis time constants, the ideal **Start Torque Constant** K_T and the **Ideal Back EMF Constant** K_E.

6. In the **Solutions** window, change the Data selection to **No-Load Operation**.

   This part displays the magnetic flux densities in the teeth and the yoke of the stator, and the yoke of the rotor. The maximum value among the three magnetic flux densities is *1.59* Tesla, which locates at the knee part of the B-H curve, below the saturation situation.

   The mmfs of the teeth and the yoke of the stator, the air-gap, the yoke and the permanent magnet of the rotor are given respectively for half magnetic reluctance path.

   The armature reaction mmf due to the armature current is referred to the demagnetization mmf. The magnetic flux leakage coefficient takes into account the part of the magnetic flux in the rotor not linking with the stator. The correction factors for the yoke lengths of the stator and the rotor to calculate the yoke mmfs of the stator and the rotor are also given here.

   The no-load revolution speed of this machine is approximately *1946* rpm.

7. In the Solutions window, change the Data selection to **Full Load Operation**.

   At **Rated Output Power (kW):***0.550*, the following characteristic parameters of the machine are calculated as:

Parameters	Calculated Values	Units
Average Input Current	*2.93*	A
(of input current waveform in one voltage period)		
RMS Armature Current	*2.45*	A
(of phase current waveform in one voltage period)		
Armature Thermal Load	*70.88*	A^2/mm^3
(product of **Specific Electric Loading** and **Armature Current Density**		)
Specific Electric Load	*14.71*	A/mm
(stator current distribution per circumferential length along air-gap)		

Armature Current Density	*4.65*	A/mm^2
(through cross-sectional area of stator wire)		
Frictional and Wind Loss	*11.42*	W
(at computed **Rated Speed**)		
Iron-Core Loss	*20.02*	W
(due to loss curves of stator and rotor iron-core materials)		
Armature Copper Loss	*47.92*	W
(stator winding ohmic loss)		
Transistor Loss	*9.14*	W
(transistor switching loss)		
Diode Loss	*0.60*	W
(diode power consumption)		
Total Loss	*89.1*	W
(sum of above losses)		
Output Power	*550*	W
(the rated operating point is derived based on **Output Power**)		
Input Power	*639.2*	W
(product of **Rated Voltage** and **Average Input Current**)		
Efficiency	*86.1*	%
(ratio of **Output Power** to **Input Power**)		
Rated Speed	*1557*	rpm
(at **Rated Output Power**)		
Rated Torque	*3.37*	Nm
(at **Rated Output Power**)		
Locked-Rotor Torque	*32.3*	Nm
(starting torque at zero revolution speed)		
Locked-Rotor Current	*52.1*	A
(starting current at zero revolution speed)		

8. In the **Solutions** window, select the **Design Sheet** tab, and scroll down to **Winding Arrangement**.

   This is the layout and the arrangement of the whole two-phase winding of phases A and B, and the short coil pitch factor *5* is taken into account. The 2-phase, 2-layer winding can be arranged in 6 slots as below:

AAABBB

Angle per slot (elec. degrees):	30
Phase-A axis (elec. degrees):	105
First slot center (elec. degrees):	0

9. In the **Solutions** window with the Design Sheet tab selected, scroll down to Transient FEA Input Data. (This is at the very bottom.)

   The following data of the armature winding corresponds to one phase armature winding.

Number of Turns	*360*	
(total number of turns viewed into output terminals)		
Parallel Branches	*1*	
Terminal Resistance	*4.13*	Ohm
(stator winding dc resistance under given operating temperature, 75^{o}C)		
End Leakage Inductance	*1.3*	mH
(of stator winding)		

   The following data is the equivalent values used to 2D electromagnetic field analyses.

Equivalent Model Depth	*65*	mm
Equivalent Stator Stacking Factor	*0.95*	
Equivalent Rotor Stacking Factor	*0.95*	
Equivalent B_r (residual flux density)	*0.87*	Tesla
Equivalent H_c (coercive force)	*690*	kA/m

10. In the **Solutions** window, click the **Curves** tab.

    This displays the Input DC Current Versus Speed graph. If the text is too small to read, you can resize the window. You can view other predefined graphs by selecting from the drop down menu in the Name field.

    Selecting the Curves tab lets you view predefined graphs for the following relations:

    - Input DC Current vs Speed
    - Efficiency vs Speed
    - Ratio of air-gap torque to DC current vs Speed
    - Output Power vs Speed
    - Output Torque vs Speed
    - Cogging Torque in Two Teeth
    - Induced Coil Voltages at Rated Speed

- Air-Gap Flux Density
- Induced Winding Phase Voltage at Rated Speed
- Winding Currents Under Load
- Phase Voltage Under Load

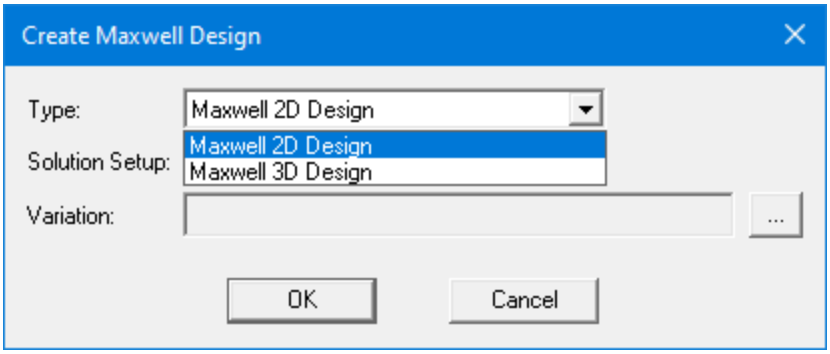
You can also create additional plots with multiple curves.

11. For example, click **RMxprt>Results>Create RMxprt Report>Rectangular Plot**.
    This displays the **Report** dialog box.
12. In the **Traces** tab, for the **Current** category, select **InputDCCurrent**, and click the **New Report** button to generate the report.
13. Then for the **Torque** category select **Output Torque** and click **Add Trace** to add the trace to the report. Click **Close** to close the **Report** window and display the combined graph.

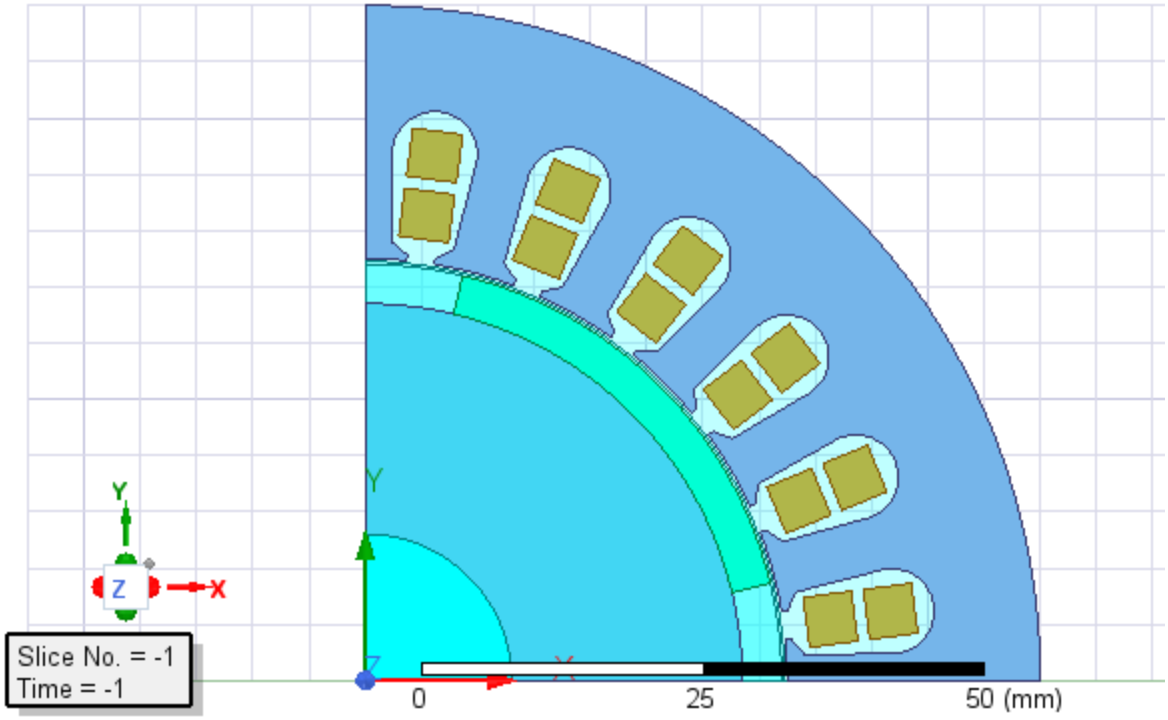
To continue to part Six of the example, go to Output Design Data.

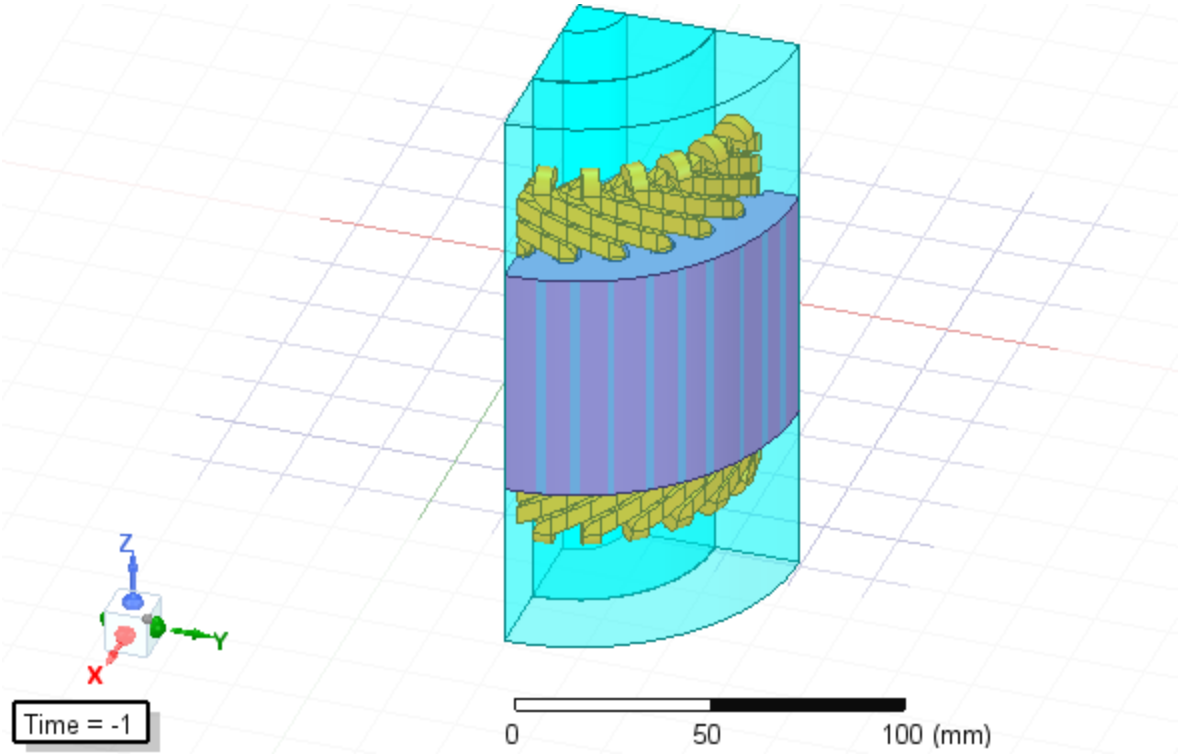
RMxprt Example Part 6: Output Design Data for Further Analysis

To analyze the model further: click **RMxprt>Analysis Setup>Create Maxwell Design**.



In the example design there is only one **Solution Setup** and no design **Variation**. For **Type** choose to create either a **Maxwell 2D Design**, or a **Maxwell 3D Design**. Click **OK** to create the Maxwell design. Examples of the resulting 2D and 3D designs are shown below.





Note	You can also export the model to a Simplorer model, a MotorCAD™project, or

	Customized Design Sheet.

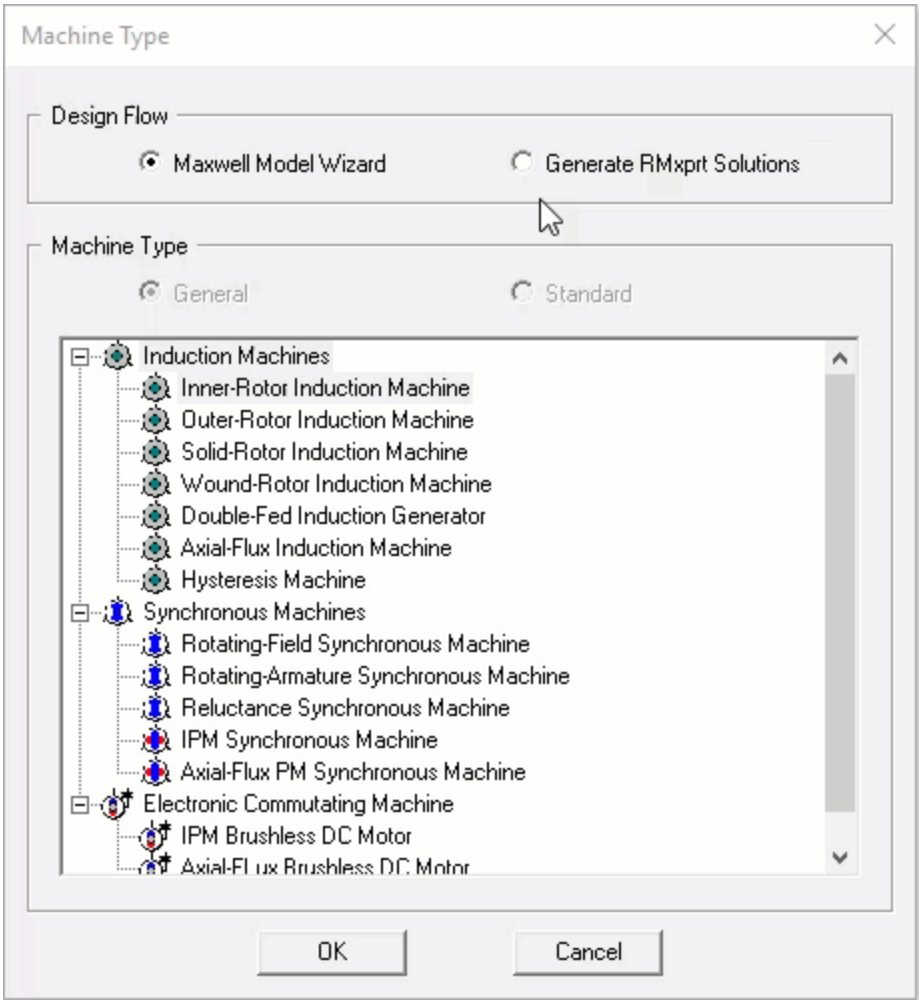
Setting Up RMxprt Projects

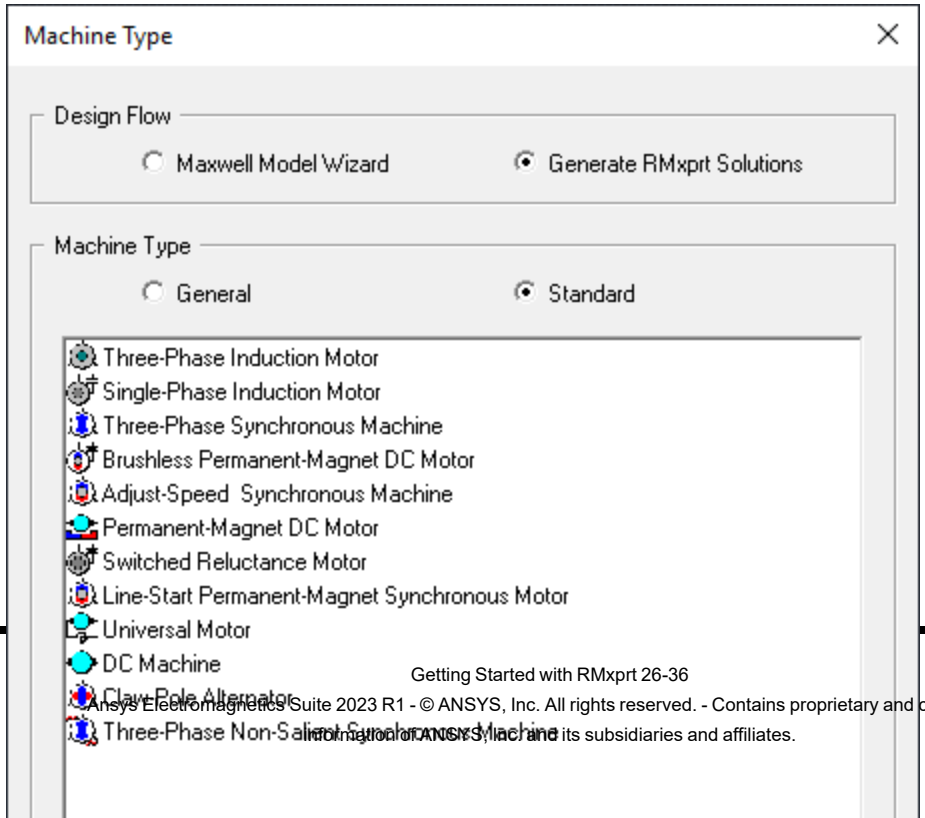
An RMxprt project is a folder that includes one or more models, or *designs*. Each design ultimately includes a geometric model, material assignments, and field solution and post-processing information.

A new project called *Projectn* is automatically created when the software is launched, where *n* is a number. You can also open a new project by clicking **File>New**. In general, use the **File** menu commands to manage projects. If you move or change the names of files without using these commands, the software may not be able to find information necessary to solve the model.

To insert an RMxprt design:

1. Choose the desired **Design Flow** option. You can select either **Maxwell Model Wizard** or **Generate RMxprt Solutions**. If **Maxwell Model Wizard** is selected, you can create an RMxprt design aimed for Maxwell design creation only. You can import your own stator and rotor geometry, with or without material assignment. If **Generate RMxprt Solutions** is selected, you can select either **General** for various pre-defined generic rotating machine types, or **Standard** for other pre-defined machine types.





2. Right-click on the design tree machine item, or pull down the RMxprt main menu, and click on **Machine Type** to open the **Machine Type<*project_name*> - <*design_name*>** dialog box.

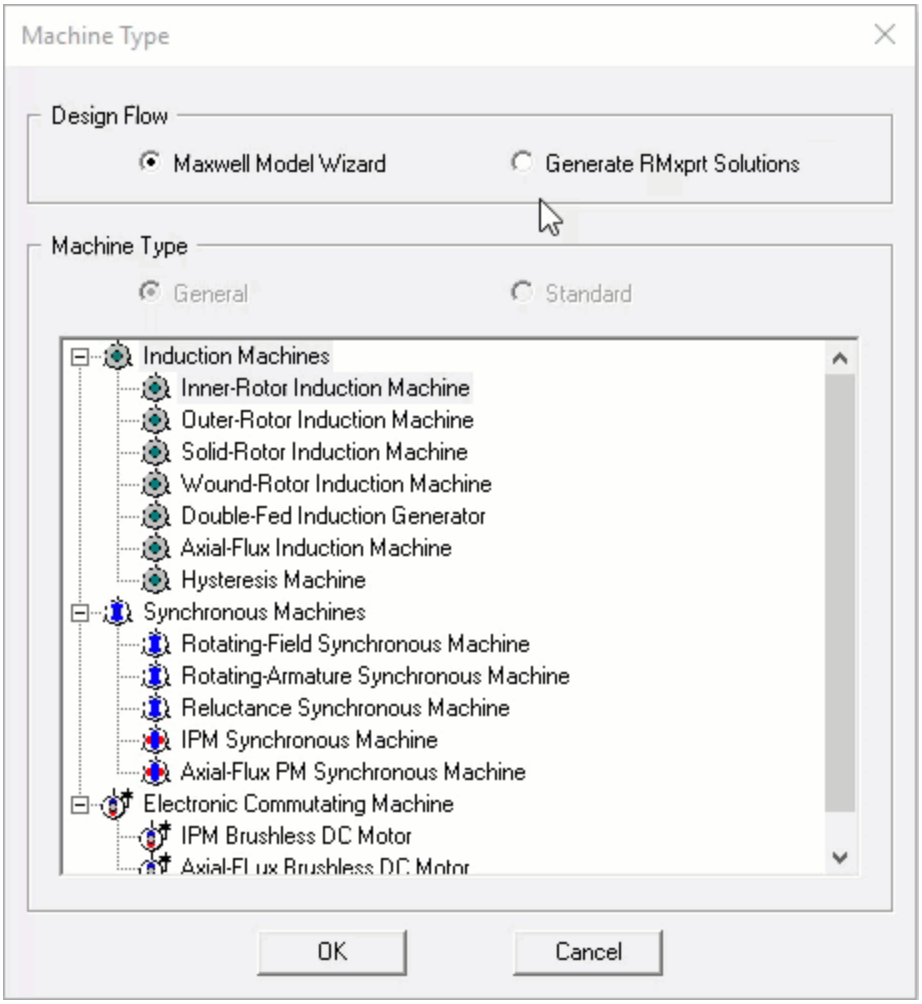
Note	The design flow cannot be changed once it's set, however the machine type can be changed for **Generate RMxprt solutions**.

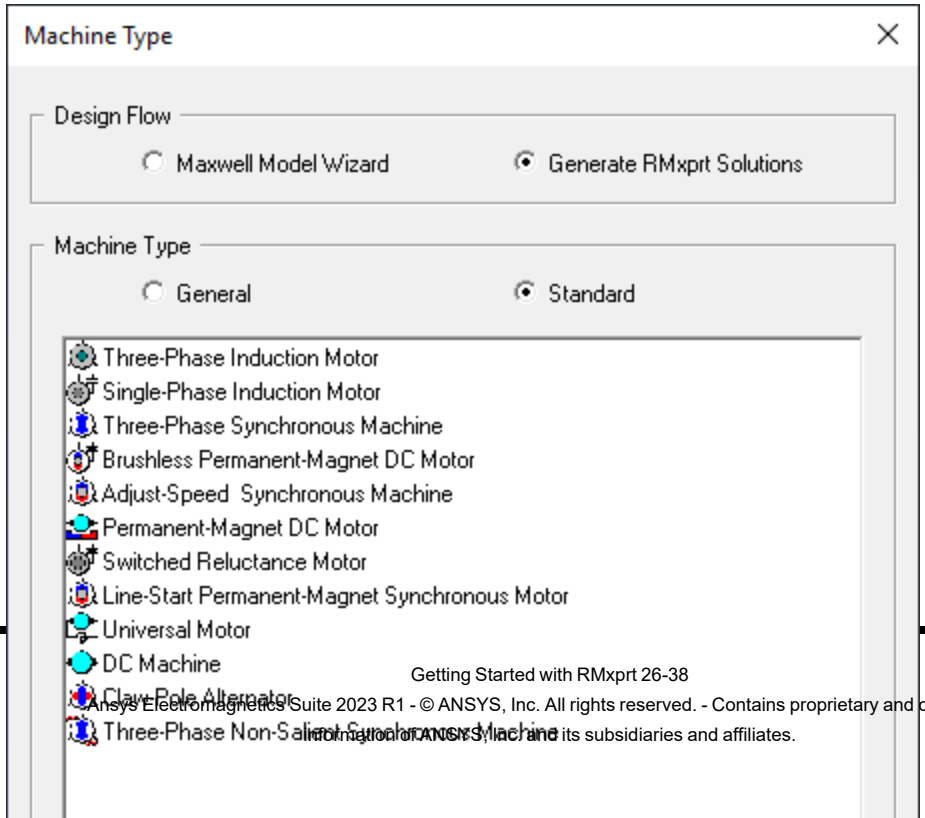
Setting Up A Machine Model

To set up an RMxprt model, follow this general procedure:

1. Insert an RMxprt design. (Click **Project>Insert RMxprt Design**, and specify the machine type from the **Machine Type** window.) For the **Design Flow**, choose either **Maxwell Model Wizard** and select a **Machine Type**; or **Generate RMxprt Solutions**.

   Generate RMxprt Solutions enables the **General** and **Standard** radio buttons. **General** allows you to pick the desired **Generic Rotating Machine** (**GRM**) machine type from the expandable list. **Standard** changes the listing to the basic RMxprt machine templates.





2. Use the **Tools** menu commands to specify general options (such as post-processing and auto-save settings), solver options (such as the default process priority), and specific RMxprt options. Also specify the Machine options (such as the units and the wire setting such as the wire shape and gauge).
3. Double-click the Machine items in the project tree, to specify the settings for the various parts of the selected machine parameters.
4. Under Definitions in the project tree, assign any Materials to the machine parts, setting values such as:
   - Permanent magnet definition, including the coercivity, energy density, and relative recovery permeability.
   - BH-curve parameters.
5. Use the **Setup** commands (either on the RMxprt menu or on the Analysis or Optimetrics submenus via the project tree) to specify variable, parametric, and optimization settings.
6. Use the Validate command to validate the design.
7. Use the Analyze commands to generate a solution, run a parametric analysis, or run an optimization.
8. Use the Results post-processing commands to display the lamination and plot the solutions.

Related Topics

Specifying RMxprt Winding Data

Quick Start for RMxprt

Maxwell Model Wizard

For Maxwell Model Wizard design flow, specify the imported geometry (if any) information through the **Machine** property window to let RMxprt know what parts and attributes to generate in the Maxwell design. Then, click **Create Maxwell Design** in the analysis setup menu to create a new Maxwell design.

You can also change machine type, and design flow in **Maxwell Model Wizard**.

Use the **Machine** property window to describe the general properties and geometry import information of the machine model to be created. It consists of the following:

Name	Description
Source Type	AC or DC source to deliver electrical power
Inner Rotor	Inner rotor, outer rotor, or axial-flux rotor
Use Minimum Setting	Hide input settings not needed for basic Maxwell model creation. Use this option to simplify the RMxprt design process to create the Maxwell model (design).
Import Option	Import geometry or Maxwell design (for geometry with material assignment).

	None: Will not import external geometry file for stator/rotor. **Geometry only**: An external geometry file needs to be imported to the design. Supports third party geometry file as well as Maxwell design(.aedt). RMxprt will add material assignments as well as other Maxwell design setups. **Maxwell design**: Only supports import of Maxwell design. The material of geometry will be retained. RMxprt will not add or change material assignments of the imported geometry.
Core Type	The stator/rotor core type
Core Alignment	Use this option to determine the alignment position of core center. Used for RMxprt to calculate the positions of stator/rotor parts, such as magnets, windings, etc. **Pole or tooth center aligned with x-axis**: When the pole center or tooth center of user's geometry is aligned with x-axis, user should choose this option. **Inter-pole or slot center aligned with x-axis**: When the inter-pole or slot center of user's geometry is aligned with x-axis, user should choose this option. Otherwise, adjust the geometry and then choose either option to prevent incorrect Maxwell design geometry generation.
Import Geometry	The geometry file to be imported. **Note** The imported geometry must be the same dimension as the specified Geometry Mode.
Geometry Mode	Specify the geometry dimension (**2D** or **3D**) so that RMxprt can create other geometry parts of same dimension.
Import Alignment	This option will only be available when fractions value is greater than 1. **Geometry center on x-axis**: When the fraction center is aligned with x-axis, choose this option to let RMxprt know the import alignment. **Independent boundary on x-axis**: When edge (or independent boundary) is aligned with x-axis, choose this option to let RMxprt know the import alignment. Otherwise, adjust the geometry and then choose either option to prevent incorrect Maxwell geometry generation.
Import Fractions	Number of fractions of the import geometry. When the value of **Import Fractions** is 1, the created geometry will have the same fractions as the traditional design without importing cores; when the value of **Import Fractions** is greater than 1, the created geometry will have the same fractions as the Import Fractions.
Z Axis Fractions	Only used when geometry mode is 3D.

	Whole: The center of the geometry model is located at the center of the coordinate and symmetric along x-y plane. **Half**: The geometry only includes half of the motor machine. All parts of the geometry model are located at the positive z axis.
View/Edit Geometry Model	Click this button to load the geometry to be imported and view/edit the content of the geometry.

The following table shows the design settings under different import combinations:

	Import geometry stator core only	Import geometry stator core with material assignments (Maxwell design)	No stator core import
Import geometry rotor core only	Rotor: material assignments and other rotor parts. Stator: material assignments and other stator parts.	Rotor: material assignments and other rotor parts. Stator: other stator parts.	Rotor: material assignments and other rotor parts. Stator: stator core with material assignments and other stator parts.
Import geometry rotor core with material assignments (Maxwell design)	Rotor: other rotor parts. Stator: material assignments and other stator parts.	Rotor: other rotor parts. Stator: material assignments and other stator parts.	Rotor: other rotor parts. Stator: stator core with material assignments and other stator parts.
No rotor core import	Rotor: rotor core with material assignments and other rotor parts. Stator: material assignments and other stator parts.	Rotor: rotor core with material assignments and other rotor parts. Stator: other stator parts.	Rotor: rotor core with material assignments and other rotor parts. Stator: stator core with material assignments and other stator parts.

Related Topics

RMxprt Machine Types
Creating Maxwell Model

Creating Maxwell Model

To generate a Maxwell model:

1. Create or select a solution setup in the project tree.
2. Right-click, and select **Create Maxwell Design** if there is any imported geometry. The geometry mode (2D or 3D) of generated Maxwell design will depend on the your **Geometry Mode** entered in the Machine properties. A Maxwell geometry model is created.
3. If there is no imported geometry, you can select either **Create Maxwell 2D Design** or **Create Maxwell 3D Design** to specify the desired geometry mode to be created.

Changing the Machine Type

RMxprt allows you to change the machine type for an existing design. The original machine data which applies to the new machine type is not retained. Instead, the initial default data for the new machine type is used.

To change the machine type for an existing model:

1. Right-click on the design tree machine item, or pull down the RMxprt main menu, and click on **Machine Type** to open the **Machine Type<*project_name*> - <*design_name*>** dialog box.
2. Choose the desired new machine type and click the **OK** button.
   - The design's machine type label is changed to that of the new machine type. For example, if the original design name and type was **3hp (Single Phase Induction Motor)**, and you change the machine type to a Switched Reluctance Motor, the design name would be **3hp (Switched Reluctance Motor)**. You can **Undo/Redo** the machine type change if you wish to revert to the original machine type and vice-versa.
   - The design tree items (such as the Machine Rotor, Stator, and Shaft data) under the design type will be updated with the default machine data applicable to the new machine type.

Related Topics

RMxprt Machine Types
Maxwell Model Wizard

SetMachineType

Use: Modifies an existing machine type.

Command: **RMxprt>Machine Type** or right-click on a machine model in the Project Manager and select **Machine Type** on the context menu.

Syntax: SetMachineType <MachineType>

Return Value: None

Parameters: <MachineType>

Type: <string>

The desired machine type.

Possible values are: “ASSM”, “BLDC”, “CPSM”, “DCM”, “GRM”, “LSSM”, “PMDC”, “SPIM”, “SRM”, “TPIM”, “NSSM”, “TPSM”, “UNIM”.

representing respectively:

Adjust-Speed Synchronous Machine, Brushless Permanent-Magnet DC Motor, Claw-pole Synchronous Machine, DC Machine, Generic Rotating Machine, Line-Start Permanent-Magnet Synchronous Motor, Permanent-Magnet Synchronous Generator, Single-Phase Induction Motor, Switched Reluctance Motor, Three-Phase Induction Motor", Non-Salient Synchronous Machine, Three-Phase Synchronous Machine, Universal Motor

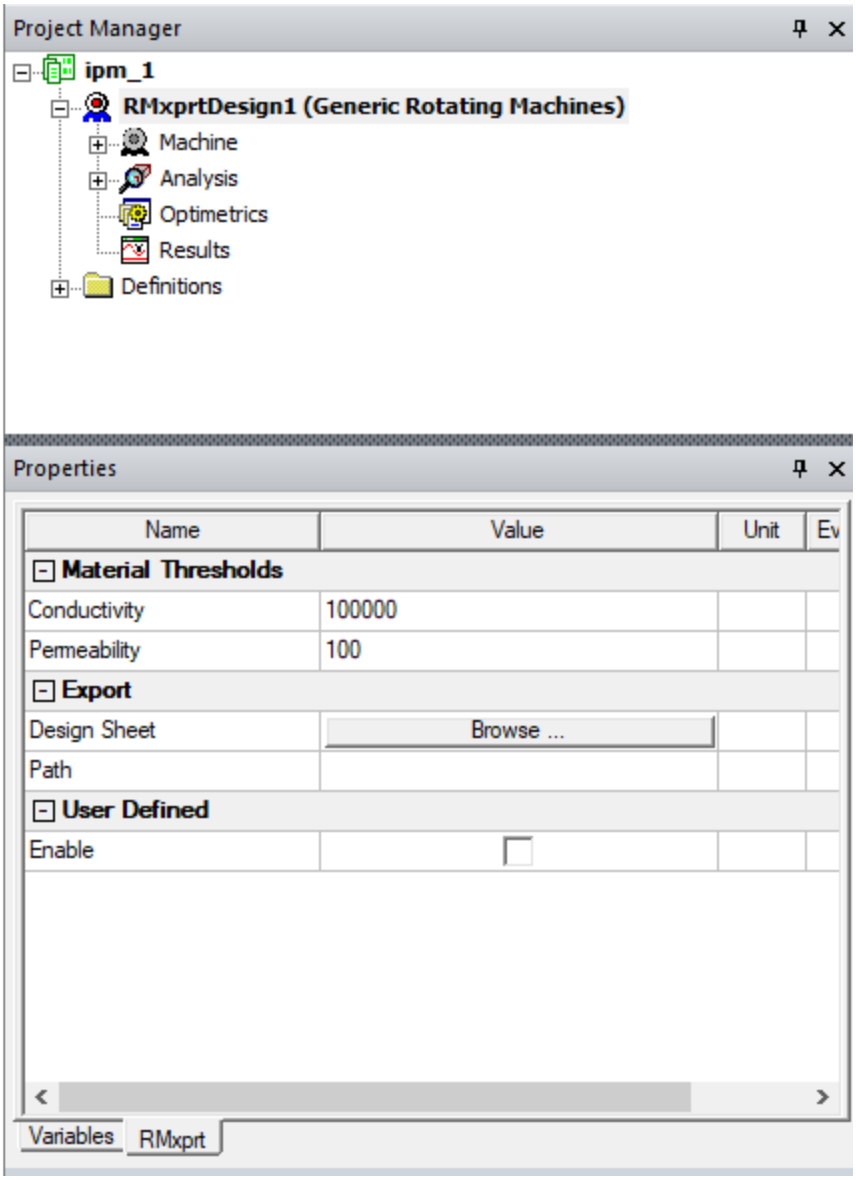
Example: SetMachineType “ASSM”

Design Settings in RMxprt

The Design Settings dialog allows you to specify how the simulator will deal with some aspects of the design.

- Set the Material Threshold for treating materials as conductors/insulators.
- Set Export Options.
- Specify User Defined Data.

You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.



Setting the Material Threshold in RMxprt

1. Click **RMxprt>Design Settings.**
   The **Design Settings** dialog box appears with the **Material Threshold** tab selected. You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.
2. Type a value in the **Conductivity Threshold** text box (Default=10,000).
3. Type a value in the **Permeability** text box (Default=100).

Note	RMxprt will treat materials with conductivity greater than 10,000 as conductors, and materials with Permeability greater than 100 as steels.

4. If you want these values to be the default, change the values by clicking the **Tools>Options>General Options** menu and setting the material thresholds in the **RMxprt>Threshold** panel.
5. Click **OK**.

Related Topics

Setting RMxprt Options

RMxprt Export Options

To set export options for the project:

1. Click **RMxprt>Design Settings**.

   The **Design Settings** dialog box appears. You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.
2. For the **Design Sheet**, type a file name in the **Excel Template** text box.
   - You can also click the **...** button to find and select a file.
3. Click **OK**.

Related Topics

Generating a Custom Design Sheet for RMxprt

Setting User-Defined Data for a Design

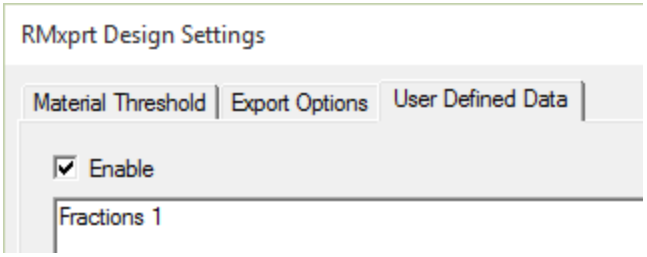
RMxprt allows a user to define some design data in a text file which can be created by a text editor, instead of by RMxprt UI, for the following special circumstances:

- Some special requests from a user which are not necessary to be added to RMxprt UI;
- Some common requests from users which have been implemented in RMxprt solver, but have not been added in RMxprt UI.
- Specifying various calibration factors.
- Specifying Data Based on Design Template Files.

When a user's requests have been implemented in an RMxprt solver but have not been added in the RMxprt UI, the updated solver and the required file format for user defined data will be sent to the user. To use the user defined data feature, the user must first edit the data file using a text editor according to the format provided. Then, select **RMxprt>Design Settings** to display the Design Settings dialog box. You can also view and set design settings by selecting the current design, and then in the **Properties** window, select the tab for the solver.

1. Select the **User Defined Data** tab.
2. Click the **Enable** check box to enable the use of **User Defined Data**.

3. User defined data may be entered directly into the text box. Click in the box and enter the data entries desired.
    - For example: If you want to create a full 360 deg model of the solved RMxprt design, enter **Fractions 1** in the text box and click **OK**. After solving the RMxprt design, if you choose to create a Maxwell 2D or 3D design, a full 360 model is generated.



Similarly, if you want to create a 180 deg half-model of the solved design, enter **Fractions 2** in the text box.

> **Note:** There are template files of file type **.temp** installed with many of the RMxprt examples located in the *<installdir>***Examples\RMxprt** folder. These text files list keywords (such as **Fractions**), default values, and descriptions of supported user defined data for their related machine types. Refer to *Specifying Data Based on Design Template Files* for details of these template files.

4. Alternatively, click **Import File** to import user defined data from an external file.
5. Browse to the directory containing the file.
6. Select the user defined data file which will be displayed in **File** name box.
7. Click **Open** to confirm the selection.
8. The file contents will be imported into the text box. Click **OK** to complete the setup.

User Defined Data is saved in the design file. Changes to **User Defined Data** will cause existing solutions to become invalid.

Related Topics

"Specifying Calibration Factors via User-Defined Data" below

"Specifying Data Based on Design Template Files" on the facing page

Specifying Calibration Factors via User-Defined Data

Calibration factors for various machine parameters are listed below. These can be used for setting user-defined data for a design.

Keyword	Default Value	Description	Applicable to
KCoreloss	1.0	Calibration factor for core loss	All machines
KE0	1.0	Calibration factor for induced voltages produced by PM or field current excitation	All synchronous machines, BLDC motors
KLad	1.0	Calibration factor for d-axis armature reactive inductance	All machines with AC armature windings
KLaq	1.0	Calibration factor for q-axis armature reactive inductance	All machines with AC armature windings
KLs_Armature	1.0	Calibration factor for armature winding slot leakage inductance	All machines with AC armature windings
KLs_Damper	1.0	Calibration factor for damper, or squirrel-cage, slot leakage inductance	All machines with damper
KLs_Field	1.0	Calibration factor for field winding slot leakage inductance	All machines with field winding
KLe_Armature	1.0	Calibration factor for armature winding end leakage inductance	All machines with AC armature windings
KLe_Damper	1.0	Calibration factor for damper, or squirrel-cage, end leakage inductance	All machines with damper
KLe_Field	1.0	Calibration factor for field winding end leakage inductance	All machines with field winding
KLend_Armature	1.0	Calibration factor for armature winding end-coil length	All machines with AC armature windings
KLend_Field	1.0	Calibration factor for field winding end-coil length	All machines with field winding

Note	
	• For field winding, the spaces between poles are referred to as slots. • Rotor winding is referred to as "Field Winding" for DFIG machines.

Related Topics

"Setting User-Defined Data for a Design" on page 26-45

Specifying Data Based on Design Template Files

There are template files of file type **.temp** installed with some of the RMxprt examples located in the <*installdir*>**Examples\RMxprt** folder. These text files list keywords, default values, and descriptions of supported user defined data for their related machine types. These can be used to set user-defined data for a design.Below are the contents of the **.temp** files located in the <*installdir*>**Examples\RMxprt** folder.

...\afpm\afbldc.temp (Axial flux brushless DC motor)

```
LeadAngle 0 // the angle that applied AC voltage leading the back
EMF
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
```

...\assm\assm.temp (Adjustable speed synchronous motor)

```
WireResistivity 0 // when WireResistivity = 0, use default value of
0.0217 ohm.mm^2/m
WireDensity 0 // when WireDensity = 0, use default value of 8900
kg/m^3
LimitedTorque 0 // when LimitedTorque < ComputedRatedTorque, use
ComputedRatedTorque for flux weakening control
ControllingIq 0 // controlling q-axis current for dq-current
control, 0 without dq-current control
ControllingId 0 // controlling d-axis current for dq-current
control
ArcBottom 0 // 0(LineBottom); 1(ArcBottom) for slot types 3 & 4
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
```

...\bldc\bldc.temp (Brushless DC motor)

```
ArcBottom 0 // 0(LineBottom); 1(ArcBottom) for slot types 3 & 4
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
```

...\cpsg\cpsg.temp (Claw pole synchronous machine)

```
Connection 0 // 0(Wye); 1(Delta) for stator winding connection
ArcBottom 0 // 0(LineBottom); 1(ArcBottom) for slot types 3 & 4
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
```

...\dcm\dcm.temp (DC machines)

```
ArcBottom 0 // 0(LineBottom); 1(ArcBottom) for slot types 3 & 4
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
LineRes 0 // additional lead wire resistance in the armature branch
```

...\dfig\dfig.temp (Double fed induction generator)

```
StatorCktType 0 // 0(Y3); 1(L3); 2(S3); 3(C2); 4(L4); 5(S4)
RotorCktType 0 // 0(Y3); 1(L3); 2(S3); 3(C2); 4(L4); 5(S4)
MPPTUntilRatedSpeed 0 // 0(Normal MPPT); 1(MPPT until rated speed,
then const wind power)
```

...\indm1\indm1.temp (Single phase induction motor)

```
SpeedAdjustMode 0 // 0(None); 1(L-Mode); 2(T-Main); 3(T-Aux)
AdjustTurnRatio 0 // turn ratio of the adjusting winding to the
original main/aux winding
AdjustDiaRatio 0 // wire diameter ratio of the adjusting winding to
the original main/aux winding
WireResistivity 0 // when WireResistivity = 0, use default value of
0.0217 ohm.mm^2/m
WireDensity 0 // when WireDensity = 0, use default value of 8900
kg/m^3
AuxResistivity 0 // when AuxResistivity = 0, use default value of
0.0217 ohm.mm^2/m
AuxDensity 0 // when AuxDensity = 0, use default value of 8900
kg/m^3
BarFitGap 0 // the fitting gap between bar and slot for squirrel-
cage rotor
StartConstrain 0 // the constraint of the ratio of the start
current or torque to the rated one
AuxCoilOnTop 0 // 0(MainOnTop); 1(AuxOnTop): for sin-wave windings
only
ArcBottom 0 // 0(LineBottom); 1(ArcBottom) for slot types 3 & 4
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
```

...indm1\3hp.temp (Single phase induction motor - 3hp)

```
SpeedAdjustMode 1 // 0(None); 1(L-Mode); 2(T-Main); 3(T-Aux)
AdjustTurnRatio 0.5
AuxCoilOnTop 0 // 0(MainOnTop); 1(AuxOnTop): for sin-wave windings
only
```

...\indm3\indm3.temp (Three phase induction motor)

```
WireResistivity 0 // when WireResistivity = 0, use default value of
0.0217 ohm.mm^2/m
WireDensity 0 // when WireDensity = 0, use default value of 8900
kg/m^3
Xe1CorrectionFactor 1 // correction factor for stator-winding end
leakage reactance
TopSpareSpace 0 // the spare space above the working coil in a
slot, between [0, 1]
BottomSpareSpace 0 // the spare space below the working coil in a
slot, between [0, 1]
LimitedTorque 0 // when LimitedTorque < ComputedRatedTorque, use
ComputedRatedTorque for flux weakening control
ArcBottom 0 // 0(LineBottom); 1(ArcBottom) for slot types 3 & 4
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
CtrlType 3 // 0(DC); 1(CCC); 2(PWM); 3(AC)
BarFitGap 0 // the fitting gap (mm) between bar and slot for
squirrel-cage rotor
```

...\indm3\yz200-6.temp (Three phase induction motor - 6 pole)

```
TopSpareSpace 0.57 // between [0, 1]
BottomSpareSpace 0 // between [0, 1]
```

...\indm3\yz200-24.temp (Three phase induction motor - 24 pole)

```
TopSpareSpace 0 // between [0, 1]
BottomSpareSpace 0.43 // between [0, 1]
```

...\ipm\ipm.temp (Interior permanent magnet machine)

```
ArcBottom 0 // 0(LineBottom); 1(ArcBottom) for slot types 3 & 4
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
SolidCore 1 // 0 for original core; 1 for solid inner rotor core
SleeveThick 0.5 // the sleeve thickness of a solid rotor rore when
SolidCore=1
HysThick 7 // thickness of the hysteresis layer when SolidCore=1
Hci 50000 // intrinsic Hc of the hysteresis loop when SolidCore=1
Br 1.6 // Br of the hysteresis loop when SolidCore=1
PartialSol 0 // 1 for not output performance data in case error
```

...\lssm\lssm.temp (Line start permanent magnet synchronous machine)

```
ArcBottom 0 // 0(LineBottom); 1(ArcBottom) for slot types 3 & 4
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
CtrlType 3 // 0(DC); 1(CCC); 2(PWM); 3(AC)
BarFitGap 0 // the fitting gap (mm) between bar and slot for
squirrel-cage rotor
```

...\nssm\nssm.temp (Non-salient synchronous machine - 3 phase)

```
ArcBottom 0 // 0(LineBottom); 1(ArcBottom) for slot types 3 & 4
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
```

...\pmdc\pmdc.temp (Permanent magnet DC motor)

```
ArcBottom 0 // 0(LineBottom); 1(ArcBottom) for slot types 3 & 4
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
```

...pmsg\pmsg.temp (Adjust speed synchronous machine)

```
CapacitivePF 0 // 0(Inductive); 1(Capacitive) for power factor
angle
ArcBottom 0 // 0(LineBottom); 1(ArcBottom) for slot types 3 & 4
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
```

...pmsg\pmsg1.temp (Adjust speed synchronous machine - generator)

```
CapacitivePF 1 // 0(Inductive); 1(Capacitive)
Fractions 0 // for 2D/3D design of all machine types. 0: default;
1: whole model
```

...\srm\srm.temp (Switched reluctance motor)

```
ArcBottom 0 // 0(LineBottom); 1(ArcBottom) for slot types 3 & 4
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
```

...\syng3\syng3.temp (Three phase synchronous machine)

```
CapacitivePF 0 // 0(Inductive); 1(Capacitive) for power factor
angle
ArcBottom 0 // 0(LineBottom); 1(ArcBottom) for slot types 3 & 4
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
BarFitGap 0 // the fitting gap between bar and slot for squirrel-
cage rotor
```

...\unim\unim.temp (Universal motor)

```
ArcBottom 0 // 0(LineBottom); 1(ArcBottom) for slot types 3 & 4
Fractions 0 // 0(MinimumModel); 1(FullModel); 2(HalfModel); 3
(ThirdModel); ...
```

Related Topics

"Setting User-Defined Data for a Design" on page 26-45

Validating RMxprt Projects

Before you run an analysis on a model, it is very important that you first perform a validation check on the project. When you perform a validation check on a project, RMxprt runs a check on all setup details of the active project to verify that the necessary steps have been completed and their parameters are reasonable.

To perform a validation check on the active project:

1. Click **RMxprt**>**Validation Check**. You can also click the **Validate** button on the **Simulation** ribbon.

   RMxprt checks the project setup, and then the **Validation Check** window appears.
2. View the results of the validation check in the **Validation Check** window.

   The following icons can appear next to an item:

	Indicates the step is complete.
	Indicates the step is incomplete.
	Indicates the step may require your attention.

3. View any messages in the **Message Manager** window.
4. If the validation check indicates that a step in your project is incomplete or incorrect, carefully review the setup details for that particular step and revise them as necessary.
5. Click **RMxprt**>**Validation Check** to run a validation check after you have revised any setup details for an incomplete or incorrect project step**.**
6. Click **Close**.

Setting General Options in RMxprt

Default settings for many of the options in RMxprt may be set through the **Tools>Options** menu. To set general options for RMxprt:

1. Click **Tools>Options>General Options**.
   The **General Options** window appears, displaying six available tabs:
   - Project Options
   - Miscellaneous Options
   - Default Units
   - Remote Analysis Options
   - WebUpdate Options
   - Desktop Performance
2. Click each tab, and make the desired selections.
3. Click **OK**.

Related Topics

Setting RMxprt Options

Configuring a Wire Specification Library

In order to be able to select wire size for your design, your need to configure a wire specification library *before* you insert a new design in your project. RMxprt provides the following libraries of wire gauge specifications based on the current widely used standards for bare copper wire gauges (including both round and rectangular wires). These files are stored in the *syslib* directory.

American.wir	File American.wir provides dimensions for all ANSI bare wires.
ANSI_SingleFilm.wir **ANSI_HeavyFilm.wir** **ANSI_TripleFilm.wir** **ANSI_QuadFilm.wir**	Files ANSI*.wir provide dimensions for bare and film insulated copper wires with single, heavy, triple and quad builds of integer and half AWG numbers.
AWG_Int_SingleFilm.wir **AWG_Int_HeavyFilm.wir** **AWG_Int_TripleFilm.wir** **AWG_Int_QuadFilm.wir**	Files AWG_Int*.wir provide dimensions of integer AWG numbers.
Chinese.wir	
IEC_R20Grade1.wir	Files IEC_R20*.wir provide

IEC_R20Grade2.wir **IEC_R20Grade3.wir** **IEC_R40Grade1.wir** **IEC_R40Grade2.wir** **IEC_R40Grade3.wir**	dimensions of enameled copper wires with Grade-1, 2, and 3 insulations of R20 series. Files IEC_R40*.wir provide dimensions of R40 series. In files IEC*.wir, the gauge numbers are equivalent to the AWG numbers according to the nominal diameters.

1. Click **Tools>Options>General Options** to open the **Options** dialog box, displaying a hierarchical list of options. Click the + next to **Machines** to show the following choice: **Wire Settings**.
2. Select the desired wire library from the drop-down list. In addition to the System Libraries choices listed above, you can also choose wire libraries you may have in either the User or Personal libraries.
3. Edit wire data as needed.

Editing Wire Data

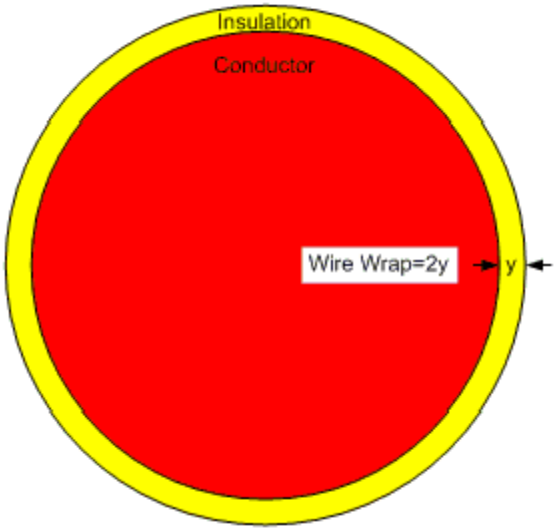
Users must generate their own data files for wire gauges using data for wire gauge and thickness of insulation provided by manufacturers. There are no national standards for insulation thickness, therefore different manufacturers produce electromagnetic wire with different thickness of insulation. The data file **American.wir** does not provide data for thickness of insulation; the data file **Chinese.wir** does provide the data for thickness of insulation, but only for the purpose of reference to users. These files are stored in the file folder *syslib* along with several other wire data reference libraries.

To define or edit wire data:

1. Click **Machine>Wire** to open the **Edit Wire Data** dialog box
2. Select the units from the **Unit System** pull-down list.

   English Unit System stands for British unit system, **Metric Unit System** stands for the metric unit system. When changing the unit system, the message box **Note** pops up to inform changing in unit system is only for specifying input data unit, but not for transferring data between two unit systems
3. Optionally, click **Import** to import wire data from a file. Provide the file name to import in the **File name:** edit box (or by browsing) and use the default file type *Wire Size File (*.wir)*. Several standard wire libraries are provided in the *syslib* directory.
4. Click the **Round** or **Rectangle** tab for the wire shape you want to edit.

   For **Round**:



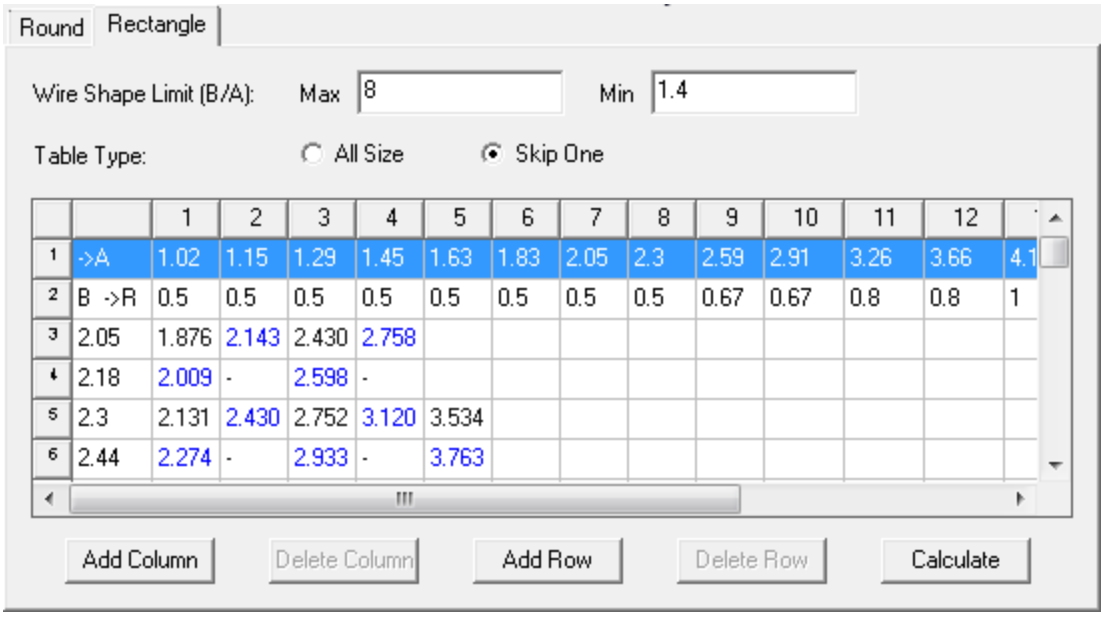
a. Specify the desired values for Gauge No., Diameter, and/or Wrap.

Gauge No.	wire gauge index number.
Diameter	diameter of bare copper wire, in mm or inch.
Wrap	thickness of insulation wrap, in mm or inch.

b. Modify the wire data as needed.

c. Optionally, Add or Delete Rows for wire values.

For **Rectangle**:



		1	2	3	4	5	6	7	8	9	10	11	12	
1	->A	1.02	1.15	1.29	1.45	1.63	1.83	2.05	2.3	2.59	2.91	3.26	3.66	4.1
2	B ->R	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.67	0.67	0.8	0.8	1
3	2.05	1.876	2.143	2.430	2.758									
4	2.18	2.009	-	2.598	-									
5	2.3	2.131	2.430	2.752	3.120	3.534								
6	2.44	2.274	-	2.933	-	3.763								

a. Specify the desired values to limit ratios of the two sides.

Wire Shape Limit (B/A) max	the maximum ratio between the wide and the narrow sides.
Wire Shape Limit (B/A) min	the minimum ratio between the wide and the narrow sides.

b. Use the radio buttons to specify whether to consider priority factors.
All Size – for No Consideration of Priority Factors

Select the radio button **All Size**, and then click the **Calculate** button. All the sectional areas of wire gauge with the ratio B/A between the wide and the narrow sides satisfying the condition **(B/A) max** > B / A > **(B/A) min** appear in the table.

Skip One – for Consideration of Priority Factors

Select the radio button **Skip One**, and then click the **Calculate** button. All the sectional areas of wire gauge with the ratio B/A between the wide and the narrow sides satisfying the condition **(B/A) max** > B / A > **(B/A) min** appear in three different modes in the table **Rectangular Wire Data**.

- At the intersection of the odd columns and the odd rows, the sectional areas appear in black numbers (recommended to use).
- At the intersection of the odd columns and the even rows or the even columns and the odd rows, the sectional areas appear in blue numbers (rarely used).
- At the cross of the even columns and the even rows, the sectional areas do not show (generally not used).

c. Optionally, Add or Delete rows or columns for wire values.

4. Optionally, click **Export** to export the data you entered to a file. Provide the file name to import in the **File name:** edit box (or by browsing) and use the default file type *Wire Size File (*.wir)*. The default directory for an exported wire data file is *userlib*.
5. When you are finished, click **Save** to save the data, and click **Close** to close the window.

Note	Saving wire data only updates the wire data in the active design.

Related Topic

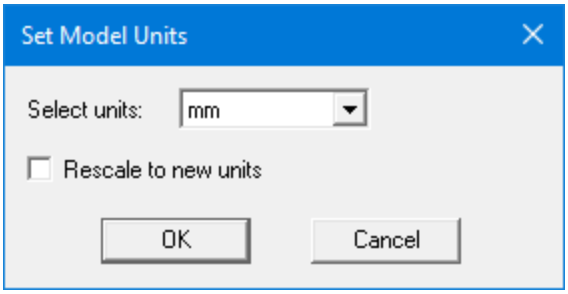
"Configuring a Wire Specification Library " on page 26-53

Setting RMxprt Machine Model Units

To set the machine model's units of measurement:

1. Click **Machine** > **Units**.

   The **Set Model Units** dialog box appears.



2. Select the new units for the model from the **Select units** drop-down menu.
3. Specify how the change in units affects the model:
   - Select the **Rescale to new units** option to rescale the dimensions to the new units. For example, selecting centimeters (cm) as the new unit of measurement results in a dimension of 10 millimeters (mm) becoming 10 cm.
   - Clear the **Rescale to new units** option (the default) to convert the dimensions to the new units without changing their scale. For example, selecting cm as the new unit of measurement results in a dimension of 10 mm becoming 1 cm.

Click **OK** to apply the new units to the model.

Edit AC Windings

RMxprt can automatically arrange almost all commonly used single- or double-layer poly-phase ac windings provided all coils have the same number of turns. Users do not need to define coils one by one. For a double-layer winding, RMxprt can also handle the coils with half turns which are arranged in the order of even, odd, even, odd, …, as long as it is physically possible.

RMxprt also provides a very flexible tool, the **Winding Editor**, to allow users to design a variety of special winding types according to their own needs, such as compound single- and double-layer

winding, big- and small-phase-spread variable-pole multiple-speed winding, sine-wave three-phase winding, and so forth. The **Winding Editor** is available to the following types of electric machines:

1. Three-phase induction motors
2. Single-phase induction motors
3. Three-phase synchronous motors and generators
4. Line-start permanent-magnet synchronous motors
5. Claw-pole alternators
6. Adjustable-speed permanent-magnet synchronous motors and generators
7. Brushless permanent-magnet DC motors

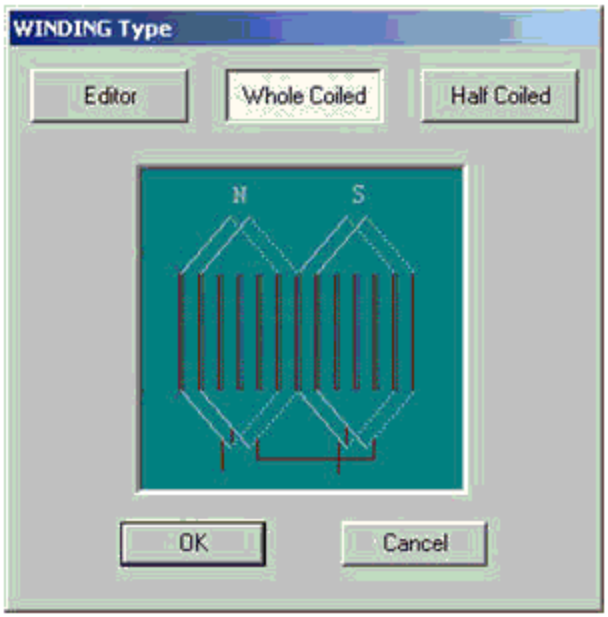
When you edit the AC winding of a new design for the first time, RMxprt creates a default winding arrangement based on the basic winding specifications: **Number of Phases**, **Number of Poles**, **Number of Slots**, **Winding Layers**, **Conductors per Slot**, and **Coil Pitch**. Then you can edit the winding configuration based on the default arrangement.

Enable Winding Editor

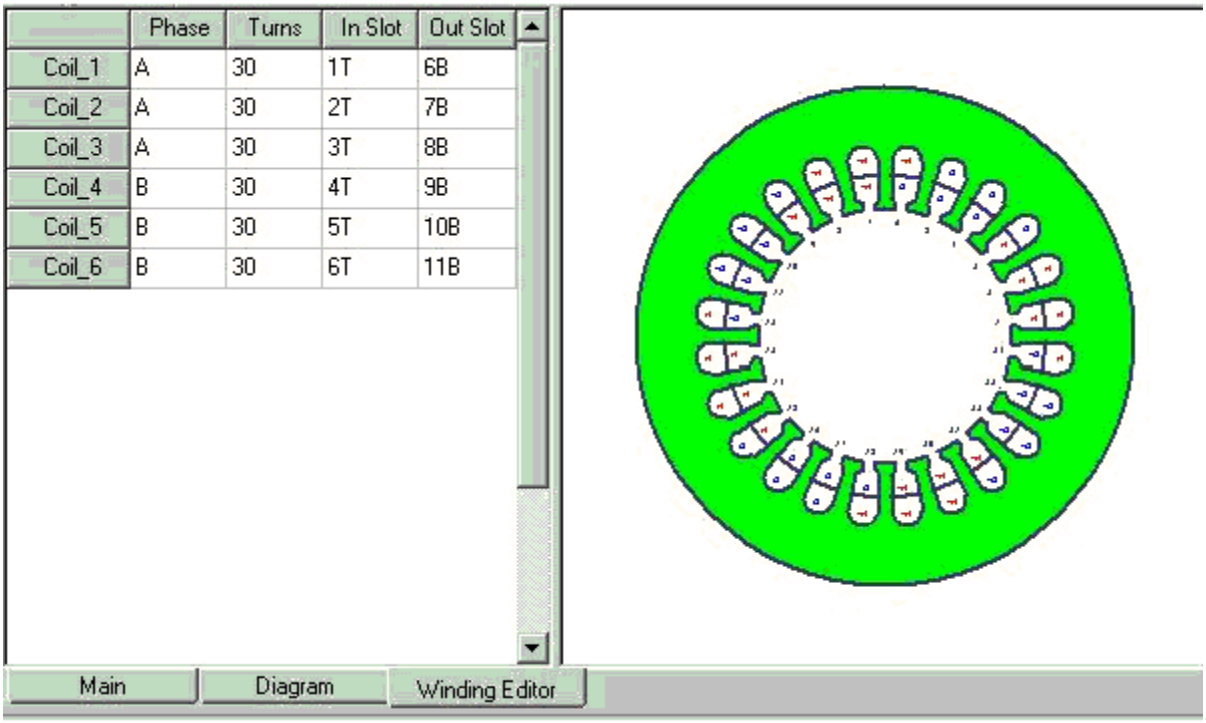
Setting the **Winding Type** property to **Editor** enables the command **Machine>Edit Layout** on the menu bar.

To display the dialog box **Winding Editor**:

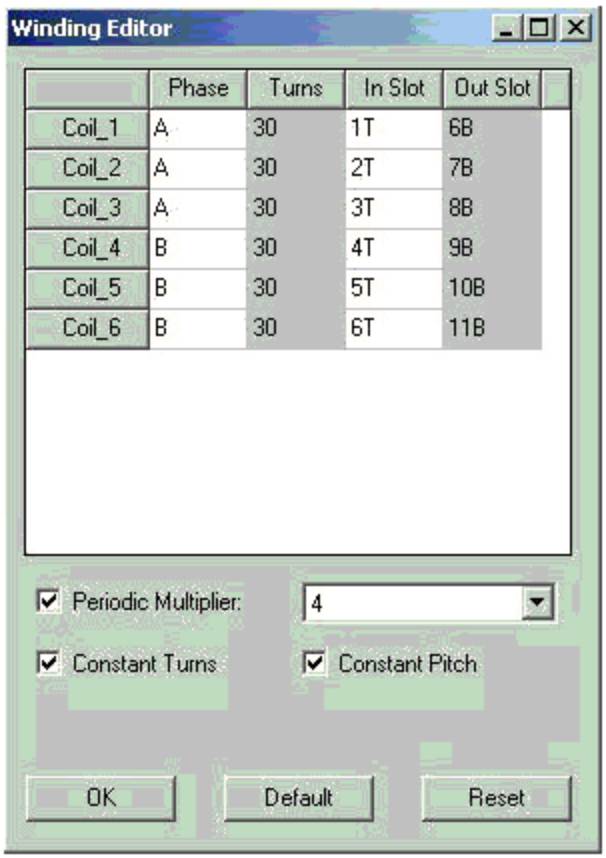
1. Select **Winding** in the Project Tree. In the Properties window, set the Winding Type Value to **Editor**. To do this, click on the button Winding Type Value to display the WINDING Type selection window, as shown:



2. Select **Editor** as the Winding Type and click **OK**.
   This closes the WINDING Type selection window and sets the Winding Type Value to Editor. It also enables the command **Machine>Winding>Edit Layout** on the menu bar. Now the Machine Editor window displays the default winding arrangement, as shown:



3. Click **Machine>Winding>Edit Layout**.
   This displays the **Winding Editor** dialog as shown. The Winding Editor dialog box includes functions that do not appear in the Winding Editor tab sheet in the RMxprt Machine Editor window.
   In addition, right-clicking in the data table section of the Winding Editor tab in the Machine Editor displays a shortcut menu where you may also select **Edit Layout**.



Edit Winding Configuration

Each row of the winding data table in the Winding Editor dialog box is identified with the coil index in the column Coil. This information is displayed in the tab sheet Winding Editor in the RMxprt Machine Editor window as well, but it is editable in the dialog box Winding Editor.

The winding data table contains four columns:

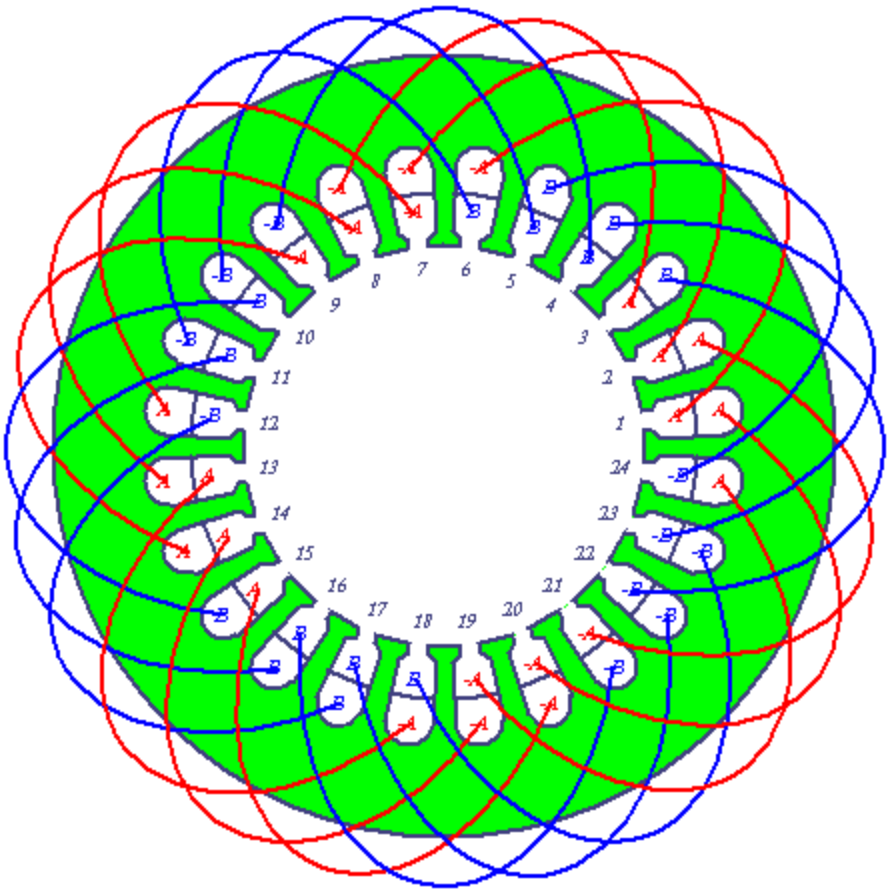
Phase	is for the phase to which the coil belongs.
Turns	is for the number of turns of the coil.
In Slots	is for the slot number with the coil side current flowing in ('flow-in-side' for short). If 2 Layers are specified in the Winding Properties window, the slot number ends with a "T" to show the top layer.
Out Slots	is for the slot number with the coil side current flowing out ("flow-out-side" for short). If 2 Layers are specified in the Winding Properties window, the slot number ends with a "B" to show the bottom layer.

View Winding Connections

When you have specified the winding data, you can execute the following commands to display or hide the winding connections.

1. Click the menu command **Machine>Winding>Connect All Coils**.

   Upon executing, the graphical display in the Machine Editor window shows the connections as shown:



2. To remove the connections in the graphical display in the Machine Editor window, select **Machine>Winding>Disconnect All Coils**.
3. Winding connections may also be viewed by shortcut menu. **Right-click** on the **winding layout** section of the Machine Editor window, a shortcut menu pops up. Select **Connect All Coils** or **Disconnect All Coils** to toggle the coils display on or off.

   If you right-click on a slot layer, commands related to that slot layer will be enabled, and you will be able to view or hide only one coil or one phase connection related to the slot layer. You may copy the connection drawing to clipboard from the shortcut menu as well.

Working with the Slot Editor

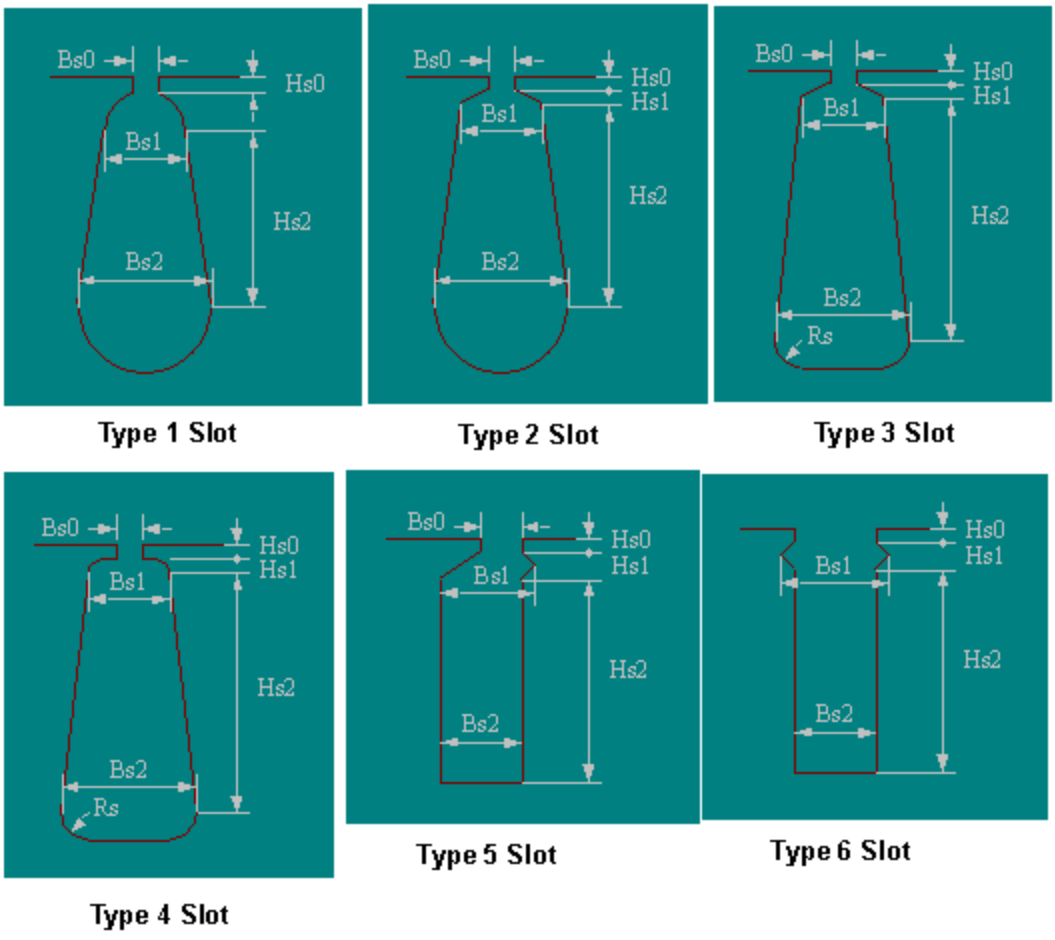
RMxprt provides a flexible tool, the **Slot Editor**, to enable users to edit user-defined slots easily.

Standard slot types in a project can be converted to equivalent user-defined versions that can be edited in the RMxprt Slot Editor tool as follows:

1. In the Project tree, select the Rotor or Stator whose slot is to be converted.
2. In the Rotor or Stator Properties Window, click the **Slot Type Value** button to open the **Select Slot Type** dialog box.
3. Check **User Defined Slot**, and click **OK**.

   The slot label in the Project tree changes to one of the following:

   UserDefSymmetricSlot (user-defined symmetric slot), **UserDefUnsymmetricSlot** (user-defined unsymmetric slot), or **UserDefHalfSlot** (user-defined half slot), depending on the standard Slot Type on which it is based.



Standard Slot Types 1, 2, 3, 4, and 6 are symmetric. Standard slot Type 5 is unsymmetric. Half slots are present if the machine's Rotor or Stator Properties Window includes a Half Slot property that has been enabled.

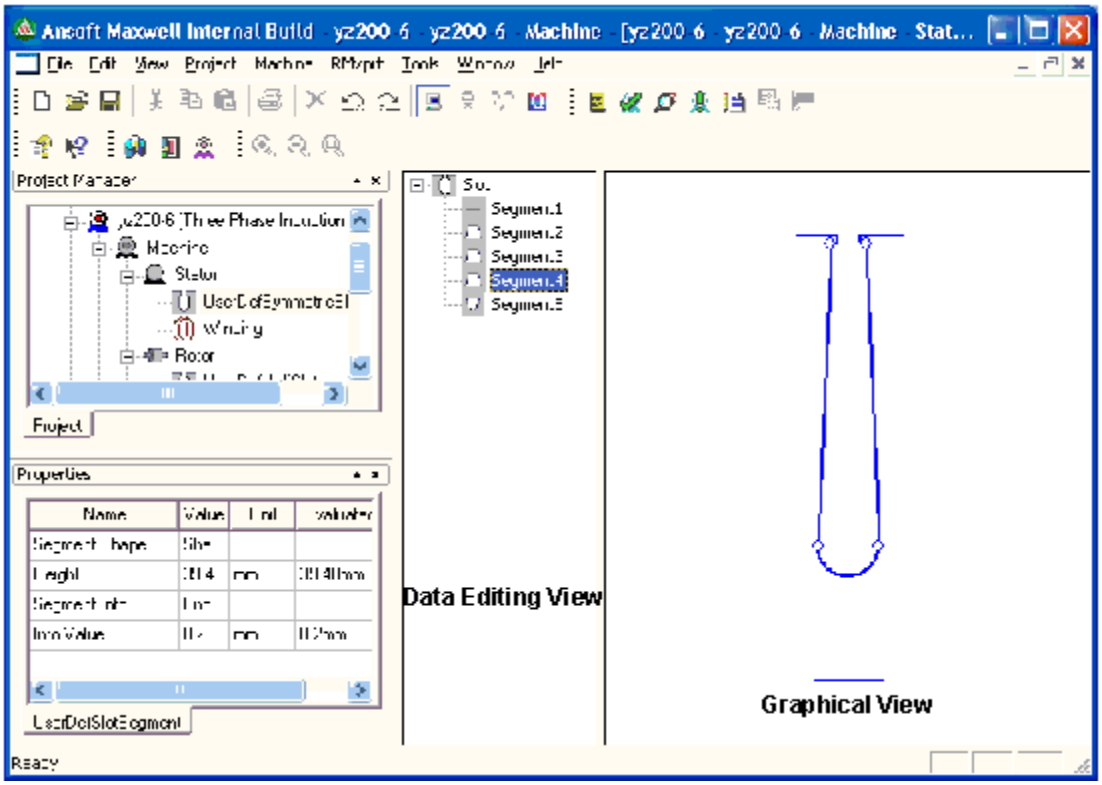
Related Topics

The Slot Editor Window
Slot Editor Data Editing View
Slot Editor Graphical View
The New Slot Dialog Box
Editing Slot Segments
Editing Unsymmetric Slots
Editing Half Slots
The Edit Slot Segments Dialog Box

The Slot Editor Window

When you select a user-editable slot in the Project tree, the Slot Editor window appears on the desktop. The slot editor window is split into two frames. The left frame is the data editing view, which contains an expandable tree view of the slot and its constituent segments. The right frame shows a graphical view of the slot geometry formed by its segments.

Slot geometry types that can be edited are: *symmetric*, *unsymmetric*, and *half-slot*. A symmetric slot is shown in the following figure.



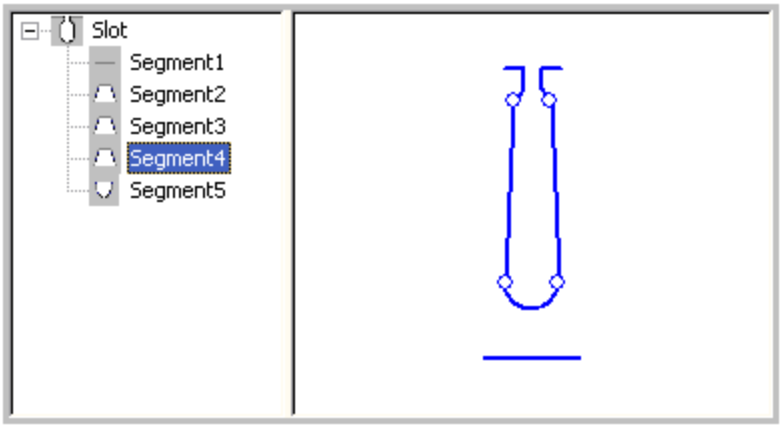
Related Topics

Slot Editor Data Editing View
Slot Editor Graphical View
The New Slot Dialog Box
Editing Slot Segments
Editing Unsymmetric Slots
Editing Half Slots
The Edit Slot Segments Dialog Box

Slot Editor Data Editing View

In the tree view, the **Slot** root tree item is predefined and cannot be modified. The top segment in the tree, **Segment1**, cannot be deleted. In the slot data edit view, the slot segment tree items always follow in order from top to bottom of the slot. Segment names also follow this top-to-bottom order, **Segment1**, **Segment2**, **Segment3**, etc., regardless of any operations that are made. Adjacent segments viewed in the tree are geometrically connected to each other as viewed on the slot graph. The data of neighboring segments are tightly coupled due to this geometry connection.

Selecting a segment in the tree highlights the corresponding segment (and its mirror image for symmetric slots) in the Slot Editor Graphic view – indicated by small open circles at the endpoints of the segments.



The Properties Window is also updated to show the selected segment's properties. Changing segment values in the slot editor updates the Properties Window, and vice-versa.

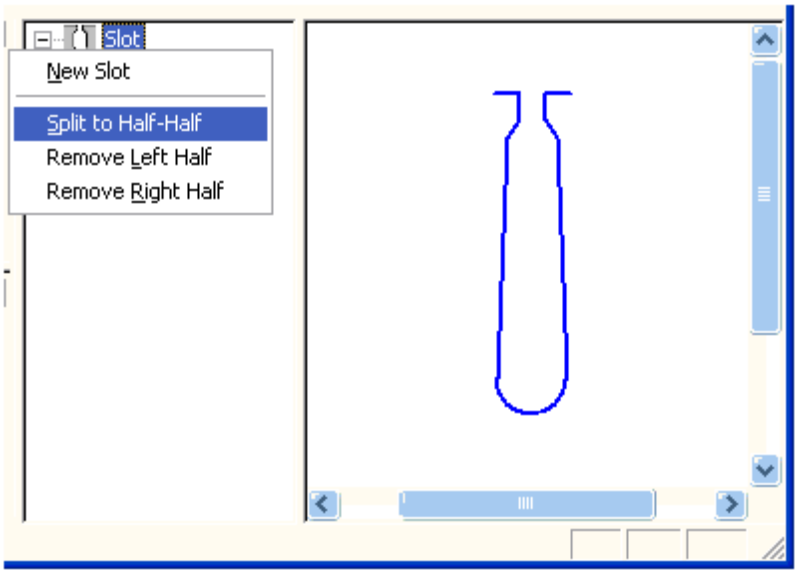
Related Topics

The Slot Editor Window
Slot Editor Graphical View

The New Slot Dialog Box
Editing Slot Segments
Editing Unsymmetric Slots
Editing Half Slots
The Edit Slot Segments Dialog Box

Editing Symmetric Slots

For a *symmetric* slot, right-clicking the **Slot** tree item pops up a context menu displaying the following choices:



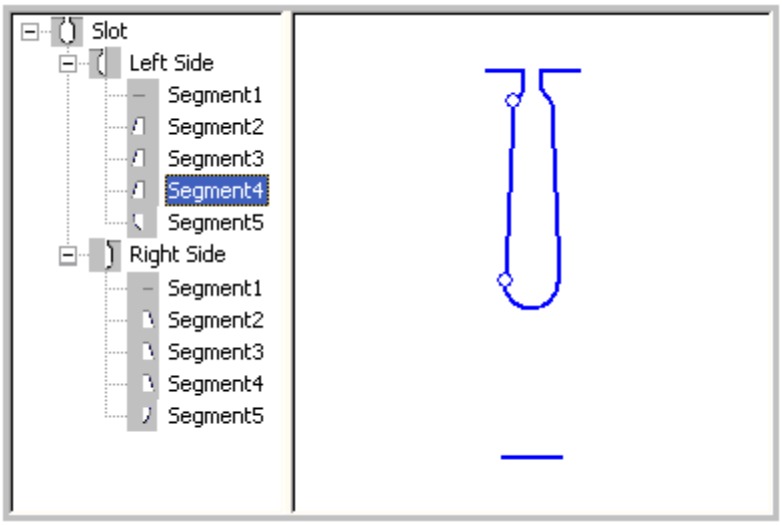
- **New Slot** - opens the **New Slot** dialog box in which the user can select a new standard slot.
- **Split to Half-Half** - splits the slot into a **Left Side** and a **Right Side** transforming it into an *unsymmetric* slot. Each side then can be edited independently.
- **Remove Left** (or **Right**) **Half**, **Remove Right Half** - removes the segments for the left (or right) half of the symmetric slot, transforming the slot into a *half-slot*.

Related Topics

The New Slot Dialog Box
Editing Slot Segments
Editing Unsymmetric Slots
Editing Half Slots

Editing Unsymmetric Slots

For an *unsymmetric* (i.e., split) slot type, two additional expandable sub-branches, **Left Side** and **Right Side**, are present in the slot tree. The segments in each side can be edited independently.



Right-clicking the **Slot** tree item pops up a context menu with the following choices:

- **New Slot** - opens the **New Slot** dialog box in which the user can select a new standard slot.
- **Merge Left** (or **Right**) **to Symmetric** - the left (or right) side segments are mirrored and merged to form a *symmetric* slot.
- **Left Right Flip** - the left side and right side segments are flipped (reflected and interchanged). The slot remains unsymmetric.
- **Remove Left Half**, **Remove Right Half** - removes the segments for the left (or right) half of the symmetric slot, transforming the slot into a *half-slot*.

Right-clicking either the **Right Side** or **Left Side** sub-branch tree item opens a context menu on which you can choose either: **Merge to Symmetric**, which mirrors the selected side's segments and merges the segments into a *symmetric* slot; or **Remove**, which removes the selected side and all of its segments, resulting in a right or left half-slot.

Related Topics

The New Slot Dialog Box

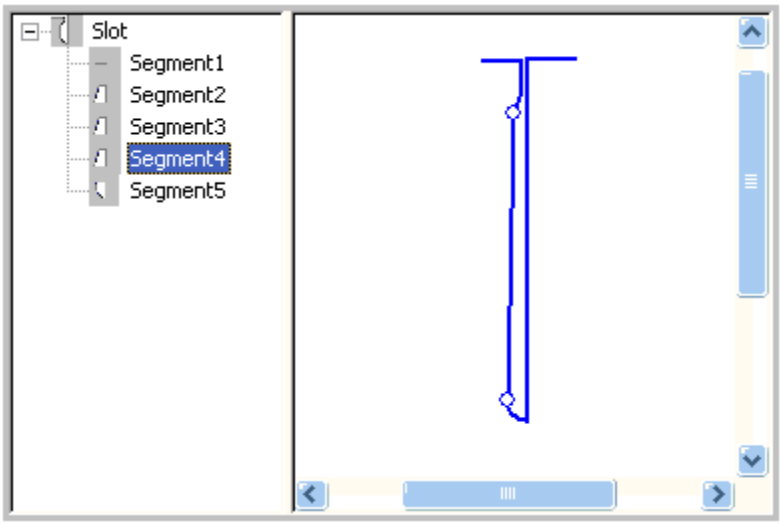
Editing Slot Segments

Editing Symmetric Slots

Editing Half Slots

Editing Half Slots

For a *half slot*, only the segments on one side of the slot can be edited.



Right-clicking the **Slot** tree item pops up a context menu displaying the following choices:

- **New Slot** - opens the **New Slot** dialog box in which the user can select a new standard slot.
- **Merge to Symmetric** - the left (or right) half-slot segments are mirrored and merged to form a symmetric slot.
- **Left Right Flip** - the left (or right) half-slot segments are flipped (reflected). The slot remains unsymmetric.
- **Mirror** - the left (or right) half-slot segments are mirrored (reflected and copied), transforming the slot into an *unsymmetric* slot.

Related Topics

The New Slot Dialog Box
Editing Slot Segments
Editing Symmetric Slots
Editing Unsymmetric Slots

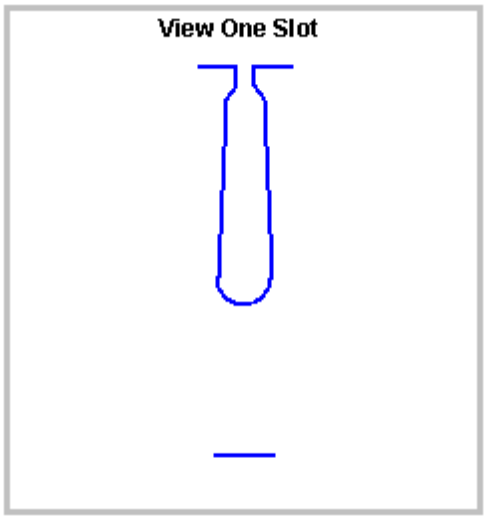
The New Slot Dialog Box

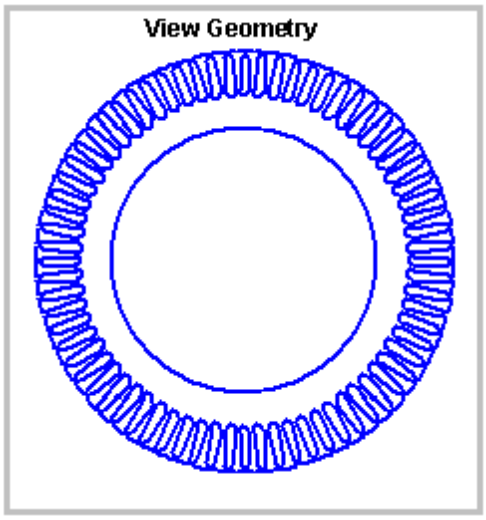
Selecting **New Slot** in any of the **Slot** tree item context menus opens the **New Slot** dialog box.

Radio buttons allow the user to select a **Symmetric Slot**, **Unsymmetrical Slot**, **Left Half Slot**, or **Right Half Slot** as the type to be added. Clicking **OK** confirms the choice and replaces the existing slot type in the editor with a slot of the selected type. The new slot has only **Segment1** present. The user must then edit the slot segments to form the desired slot geometry.

Slot Editor Graphical View

The slot editor graphic view allows users either to **View One Slot** of the type currently being edited, or to **View** the **Geometry** of the machine stator or rotor with all of the slots in place.





Right-clicking anywhere in the graphic view pops up a context menu. In addition to changing the view, the menu provides commands to **Zoom In** and **Zoom Out**, to **Fit All** the view in the frame, as well as commands to **Insert**, **Append**, **Modify**, and **Remove** slot segments. These commands are discussed more fully in the section on *Editing Slot Segments*.

Related Topics

The Slot Editor Window
Slot Editor Data Editing View
The New Slot Dialog Box
Editing Slot Segments
Editing Unsymmetric Slots
Editing Half Slots
The Edit Slot Segments Dialog Box

Editing Slot Segments

The RMxprt Slot Editor allows users to edit the segments that form the geometry of slots. In the slot data edit view, selecting a slot segment item on the tree, then right-clicking on it displays a context menu with commands that allow users to **Insert**, **Append**, **Modify**, and **Remove** segments.

Alternatively, users can select a slot segment directly in the graphic view, then right-click to bring up a context menu containing the same segment editing commands.

When a slot segment is selected, users can also modify the segment data directly in the desktop property window. Each edit triggers immediate validation. If edited values are improper, warning message windows are displayed describing the problem. Editing operations support Undo/Redo. Scripting functions are also available.

- **Insert Segment** -adds a segment *before* the currently selected segment. The added segment is defined by the user in the **Edit Slot Segment** dialog box.

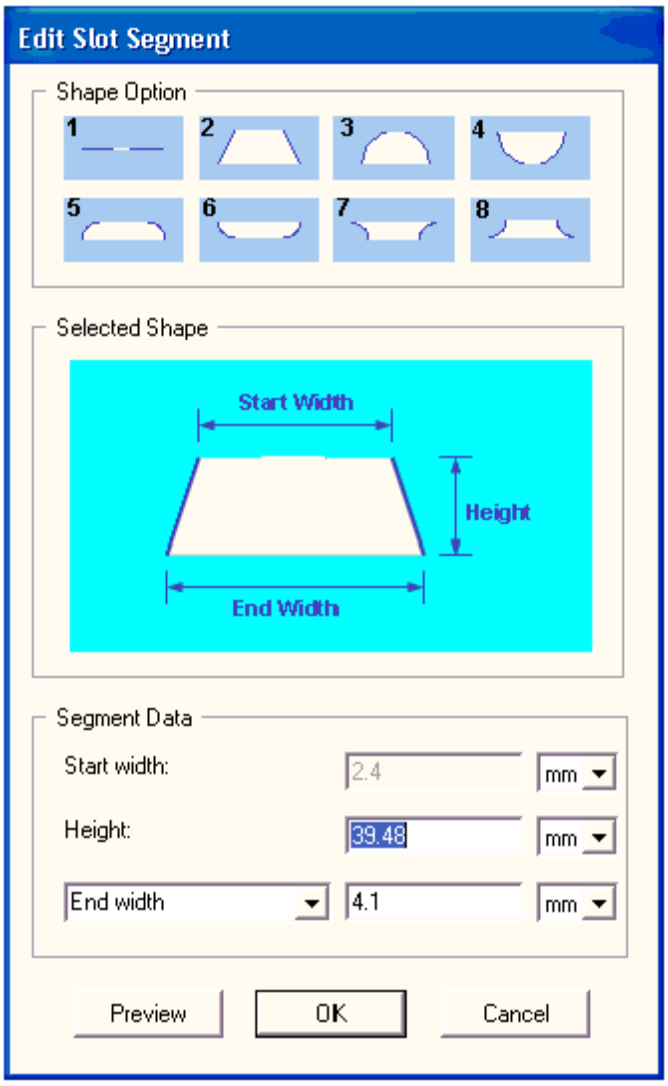
NOTE: You cannot insert a segment before **Segment1**.

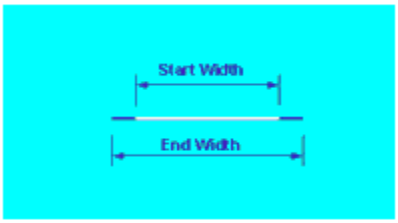
- **Append Segment** - adds a segment *after (i.e., below)* the currently selected segment. The appended segment is defined by the user in the **Edit Slot Segment** dialog box.
- **Modify Segment** - modifies the currently selected segment. The segment is modified by the user in the **Edit Slot Segment** dialog box.
- **Remove Segment** - removes the currently selected segment. The ends of the segments on either side of the removed segment are joined when the segment is removed.

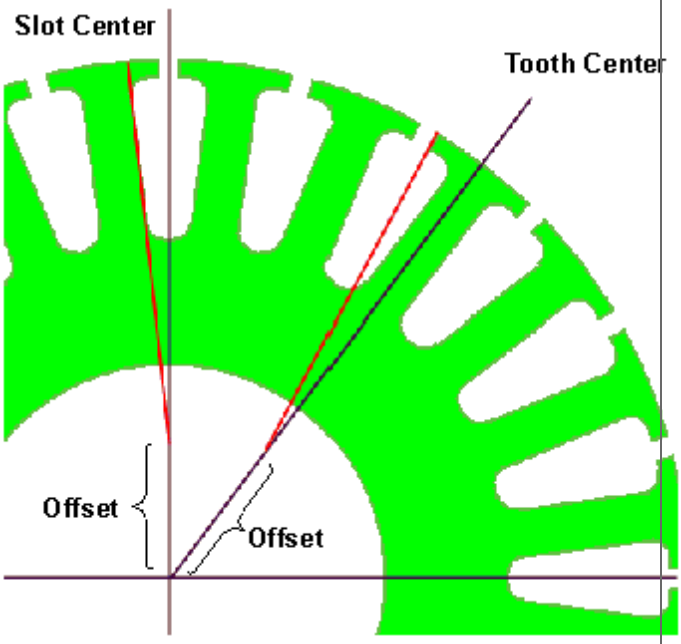
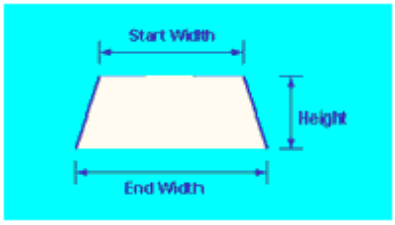
NOTE: **Segment1** cannot be removed.

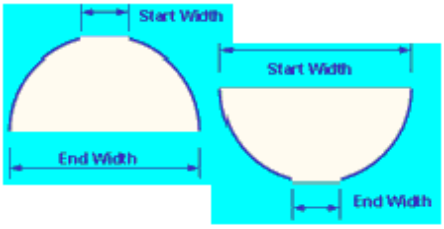
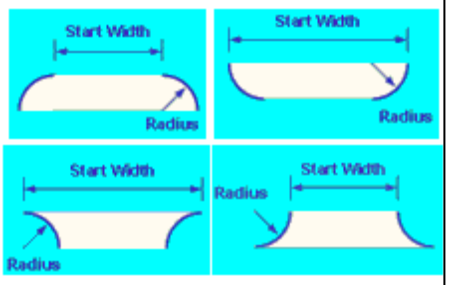
The Edit Slot Segment Dialog Box

In the **Edit Slot Segment** dialog box, user can define the shape and dimension(s) of the segment to be added, inserted, or appended relative to the currently selected segment in the slot data edit view. As shown below, eight basic geometric shapes are provided for defining the slot segment. The **Selected Shape** panel shows dimensional parameters that affect the segment shape such as: height, starting and ending width, and radius. Parameters that determine the shape of the segment can be edited in the **Segment Data** panel.



	• **Start width** - sets the starting width and unit of measure. The value shown depends on the ending width of the previous segment and thus is not editable. • **End width** - sets the ending width and unit of measure. A drop-down box provides three additional options for controlling the segment shape: • **Line edge** (the default) - makes the segment a straight line. • **Arc offset on slot center** - makes the segment an

	arc whose radius is determined by a user-specified offset from the rotor/stator center with respect to the slot center. • **Arc offset on tooth center** - makes the segment an arc whose radius is determined by a user-specified offset from the rotor/stator center with respect to the tooth center. 
	• **Start width** - sets the starting width and unit of measure. The value shown depends on the ending width of the previous segment and thus is not editable. • **Height** -sets the height and unit of measure for the segment. A drop-down box provides three options for controlling the segment shape: • **End width** (the default)- sets the ending width and unit of measure.

	•  **Parallel slot** - effectively sets the segment end width to the same value as the start width resulting in the slot sides being parallel to each other. •  **Parallel tooth** - sets the segment end width to a value such that the slot side is parallel to the adjacent slot side of the tooth formed between them.
	• **Start width** - sets the starting width and unit of measure. The value shown depends on the ending width of the previous segment and thus is not editable. • **End width** - sets the ending width and unit of measure.
	• **Start width** - sets the starting width and unit of measure. The value shown depends on the ending width of the previous segment and thus is not editable. • **Radius** - sets the radius and unit of measure for the segment.

Validation of the entered data is done when either the **OK** or **Preview** is clicked. If edited values are improper, warning message windows are displayed describing the problem. The **Preview** button allows users to preview the current changes in the slot graph window without confirming the changes.

Clicking the **OK** button confirms the changes and closes the dialog box.

Clicking the **Cancel** button cancels the changes and closes the dialog box.

Related Topics

The New Slot Dialog Box

Editing Slot Segments

Editing Symmetric Slots

Editing Unsymmetric Slots

Editing Half Slots

Working with Variables in RMxprt

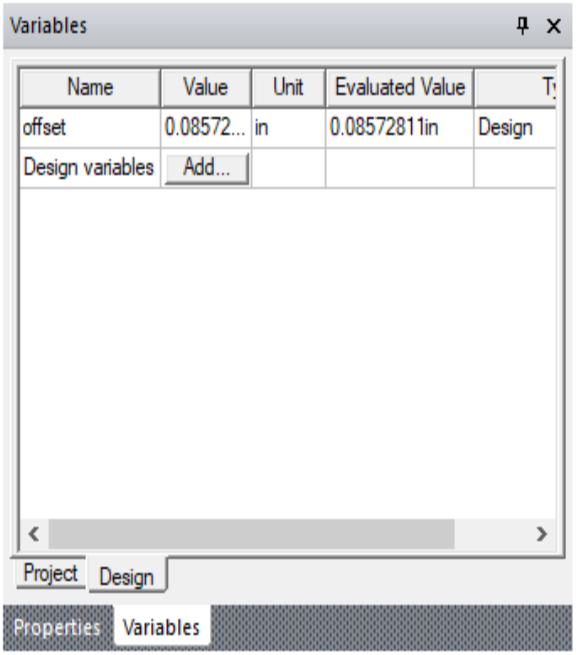
A variable is a numerical value, mathematical expression, or mathematical function that can be assigned to a design parameter in RMxprt. Variables are useful in the following situations:

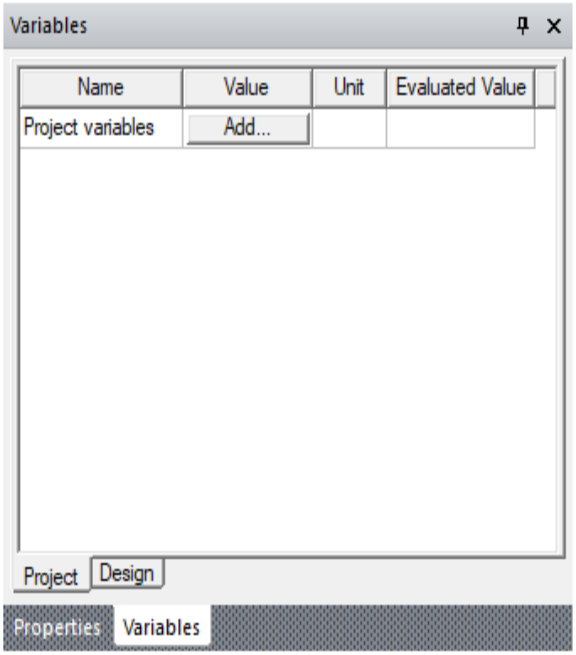
- When you expect to change a parameter often.
- When you expect to use the same parameter value often.
- When you intend to run a parametric analysis in which you specify a series of variable values within a range to solve.
- When you intend to optimize a parameter value by running an optimization analysis.

There are two types of variables in RMxprt:

Project Variables	A project variable can be assigned to any parameter value in the project in which it was created. RMxprt differentiates project variables from other types of variables by prefixing the variable name with the $ symbol. You can manually include the $ symbol in the project variable's name, or RMxprt can automatically append the $ after you define the variable.
Design Variables	A design variable can be assigned to any parameter value in the RMxprt design in which it was created.

Clicking **View** > **Variables** brings up a dockable variable window that is associated with the active project and/or design. When there is an active project, there will be a corresponding project variable tab. When there is an active design, there will be a corresponding design variable tab. Each tab contains an **Add...** button allowing creation of new variable of this type. If variables exist for the Project or Design, they are shown in the corresponding tab.





Related Topics

Setting up an Optimization Analysis

Adding a Project Variable in RMxprt

A project variable can be assigned to a parameter value in the RMxprt project in which it was created. RMxprt differentiates project variables from other types of variables by prefixing the variable name with the following symbol: **$**. You can manually include the symbol **$** in the project variable's name when you create it, or RMxprt will automatically append the project variable's name with the symbol after you define the variable. Project variables can be used in any design within the project.

1. Click **Project>Project Variables**.
   - Alternatively, right-click the project name in the project tree, and then click **Project Variables** on the shortcut menu.

   The **Properties** dialog box appears.
2. Under the **Project Variables** tab, click **Add**.

   The **Add Property** dialog box appears.
3. In the **Name** box, type the name of the variable.

   Project variable names must start with the symbol **$** followed by a letter. Variable names may include alphanumeric characters and underscores (_). The names of intrinsic functions and the predefined constant pi (π) cannot be used as variable names.

You can sort the project variables by clicking on the Name column header. By default, variables are sorted in original order. Clicking once sorts them in ascending order, noted by a triangle pointing up. Clicking against sorts in descending order, noted by a triangle pointing down. Clicking a third time sorts in original order, with no triangle.

4. Select a radio button for the variable use:

Selected Use	Setable Properties
Variable	Unit Type, Units, Value.
Separator	Value
Array Index Variable	Associate Array variable, Value

Each selection affects the settable options.

5. For Project Variables in the **Unit Type** text box you can use the drop down menu to select from the list of available unit types. “None” is the default.

   When you select a Unit Type, the choices in drop-down menu for the Units text box adapt to that unit type. For example, selecting Length as the Unit Type causes the **Unit** menu to show a range of metric and english units for length. Similarly, if you select the Unit Type as Resistance, the Units drop down lists a range of standard Ohm units.

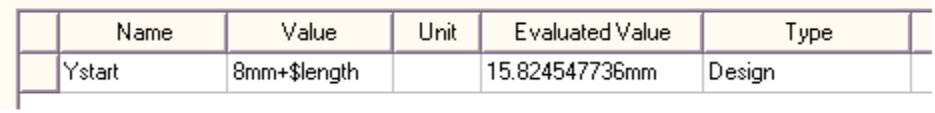
6. In the **Value** box, type the quantity that the variable represents. Optionally, include the units of measurement.

Warning	If you include the variable's units in its definition (in the **Value** box), do not include the variable's units when you enter the variable name for a parameter value.

   The quantity can be a numerical value, a mathematical expression, or a mathematical function. The quantity entered will be the *current,* or *default, value* for the variable.

7. Click **OK**.

   You return to the **Properties** dialog box. The new variable and its value are listed in the table. If the value is an expression, the evaluated value is shown. Updating the expression also changes the evaluated value display. The evaluated values of any dependent variables also are changed.



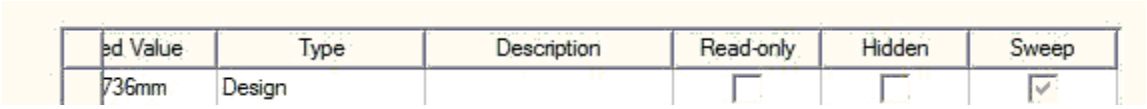
Name	Value	Unit	Evaluated Value	Type
Ystart	8mm+$length		15.82454773 6mm	Design

8. Optionally, type a description of the variable in the **Description** box.
9. Optionally, select **Read-Only**. The variable's name, value, unit, and description cannot be modified when **Read-only** is selected.

10. Optionally, select **Hidden**. If you clear the **Show Hidden** option, the hidden variable will not appear in the **Properties** dialog box.
11. You can also designate a variable as Sweep. You may need to use the scroll bar or resize the dialog to view the check boxes.

The Sweep check box lets you designate variables to include in solution indexing as a way to permit faster post processing. Variables with Sweep unchecked are not used in solution indexing. If a solution exists, checking or unchecking a variable's Sweep setting produces a warning that the change will invalidate existing solutions. To continue, click **OK** to dismiss the warning dialog.

If a variable has dependent variables, the Sweep check box is disabled and cannot be changed.



12. Click **OK**.

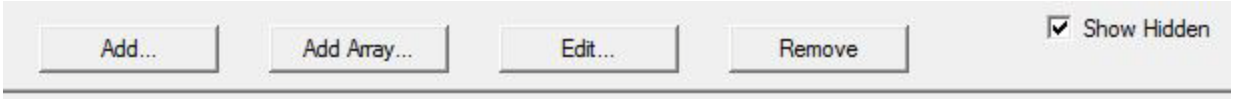
The new variable can now be assigned to a parameter value in the project in which it was created.

Adding a Design Variable in RMxprt

A design variable is associated with an RMxprt design. A design variable can be assigned to a parameter value in the RMxprt design in which it was created.

1. Click **RMxprt>Design Properties**.
   - Alternatively, right-click the design name in the project tree, and then click **Design Properties** on the shortcut menu.

   The **Properties** dialog box appears opened on the **Local Variables** tab. From the Properties dialog you can Add, Add Array, Edit, or Remove variables.



Any existing Design variables are listed in the **Properties** dialog with the name followed by cells for Value, Unit, Evaluated Value, Type, Description, and Read-only and Hidden check boxes. A Show Hidden check box on the lower right of the Properties dialog controls the appearance of any Hidden variables.

Initially, leave the radio button with Value selected until you have defined a variable. The other radio buttons let you enable defined variables for Optimization/Design of Experiments, Tuning, Sensitivity or Statistics. Selecting one of these radio buttons adds a new column to the Variable definition row for which you can check or uncheck Include for regular variables for that kind of Optimetrics simulation.

2. Click **Add**.

   The **Add Property** dialog box appears.

3. In the **Name** box, type the name of the variable.

   Variable names must start with a letter, and may include alphanumeric characters and underscores (_). The names of intrinsic functions and the predefined constant pi (π) cannot be used as variable names.

   You can sort the variables by clicking on the Name column header. By default, variables are sorted in original order. Clicking once sorts them in ascending order, noted by a triangle pointing up. Clicking against sorts in descending order, noted by a triangle pointing down. Clicking a third time sorts in original order, with no triangle.

4. Select a radio button for the variable use:

Selected Use	Setable Properties
Variable	Unit Type, Units, Value
Array Index Variable	Associate Array variable, Value
Separator	Value
Post Processing Variable	Unit Type, Units, Value

   Each selection affects the settable options.

5. In the **Unit Type** text box you can use the drop-down menu to select from the list of available unit types. “None” is the default.

   When you select a Unit Type, the choices in drop-down menu for the Units text box adapt to that unit type. For example, selecting Length as the Unit Type causes the Unit menu to show a range of metric and english units for length. Similarly, if you select the Unit Type as Resistance, the Units drop down lists a range of standard Ohm units.

6. In the **Value** box, type the quantity that the variable represents. Optionally, include the units of measurement.

Note	If you include the variable's units in its definition (in the **Value** box), do not include the variable's units when you enter the variable name for a parameter value.

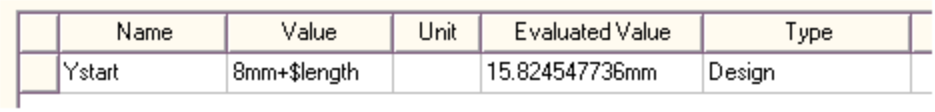
   The quantity can be a numerical value, a mathematical expression, or a mathematical function. The quantity entered will be the *current* (or *default value)* for the variable. If the mathematical expression includes a reference to an existing variable, this variable is treated as a dependent variable. The units for a dependent variable will automatically change to

those of the independent variable on which the value depends. Additionally, dependent variables, though useful in many situations, cannot be the direct subject of optimization, sensitivity analysis, tuning, or statistical analysis.

Note	Complex numbers are not allowed for variables to be used in an Optimetrics sweep, or for optimization, statistical, sensitivity or tuning setups.

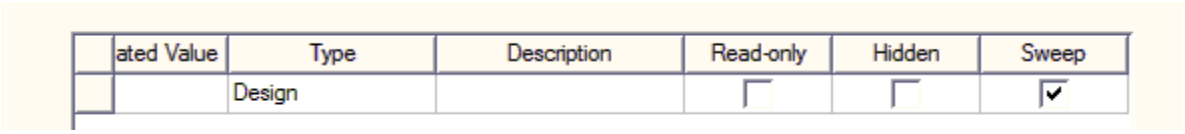
7. Click **OK**.

   You return to the **Properties** dialog box. The new variable and its value are listed in the table. If the value is an expression, the evaluated value is shown. Updating the expression also changes the evaluated value display. The evaluated values of any dependent variables also are changed.



	Name	Value	Unit	Evaluated Value	Type
	Ystart	8mm+$length		15.824547736mm	Design

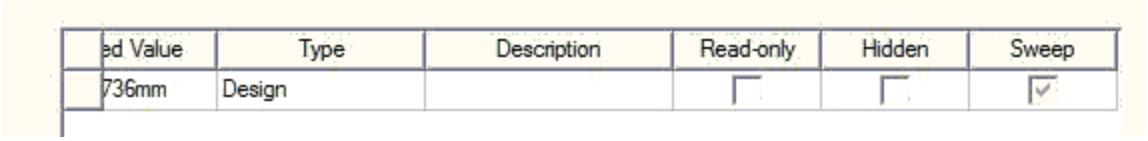
8. Optionally, type a description of the variable in the **Description** box.
9. You can designate a variable as Read-only, Hidden, or Sweep. You may need to use the scroll bar or resize the dialog to view the check boxes.



	ated Value	Type	Description	Read-only	Hidden	Sweep
		Design		☐	☐	☑

   The Sweep check box lets you designate variables to include in solution indexing as a way to permit faster post processing. Variables with Sweep unchecked are not used in solution indexing. If a solution exists, checking or unchecking a variable's Sweep setting produces a warning that the change will invalidate existing solutions. To continue, click **OK** to dismiss the warning dialog.

   If a variable has dependent variables, the Sweep check box is disabled and cannot be changed.



	ed Value	Type	Description	Read-only	Hidden	Sweep
	736mm	Design		☐	☐	☑

10. Click **OK**.

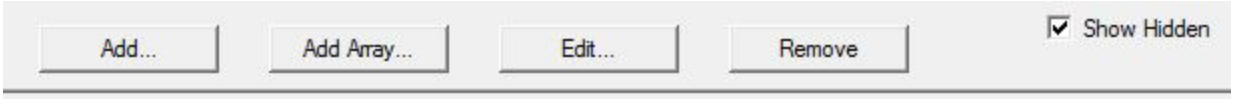
The new variable can now be assigned to a parameter value in the design in which it was created.

Add Array of Values for an RMxprt Design Variable

A design variable is associated with an RMxprt design. You can also add a variable defined with an array of values.

1. Click **RMxprt**>**Design Properties**.
   - Alternatively, right-click the design name in the project tree, and then click **Design Properties** on the shortcut menu.

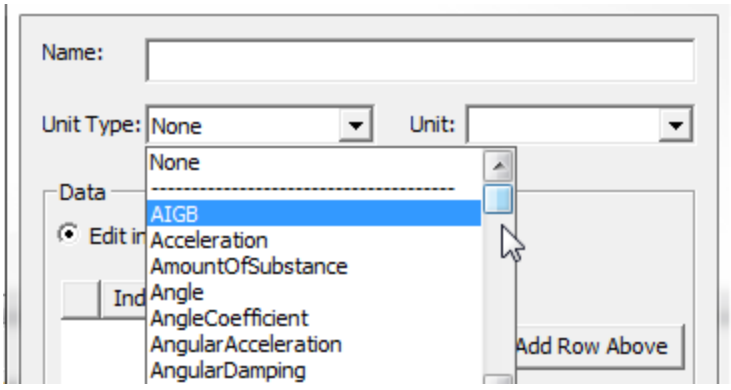
     The Local Variables **Properties** dialog box appears. From the Properties dialog you can Add, Add Array, Edit... or Remove variables. This section describes Add Array. Use the links for descriptions of Add,.Edit and Remove,



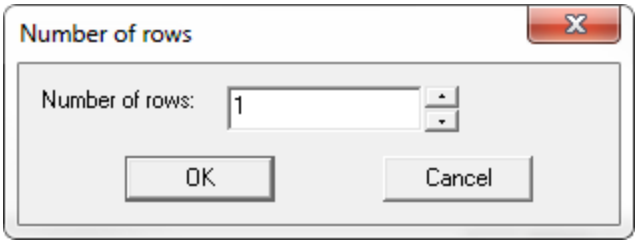
     Any existing Design variables are listed in the **Properties** dialog with the name followed by cells for Value, Unit, Evaluated Value, Type, Description, and Read-only and Hidden check boxes. A Show Hidden check box on the lower right of the Properties dialog controls the appearance of any Hidden variables.

     Initially, leave the radio button with Value selected until you have defined a variable. The other radio buttons let you enable regular variables for Optimization/Design of Experiments, Tuning, Sensitivity or Statistics. Array variables cannot be enabled for Optimetrics use.
2. Click the **Add Array...**. button.

   The **Add Array** dialog displays.
3. Specify a variable Name in the text field.
4. Select a Unit Type and Units from the drop-down menus.



5. To specify the array with Edit in Grid Selected, you can begin by clicking the **Append Rows...** button to display the **Number of Rows** dialog box. (For Edit in plain text field, see below.)



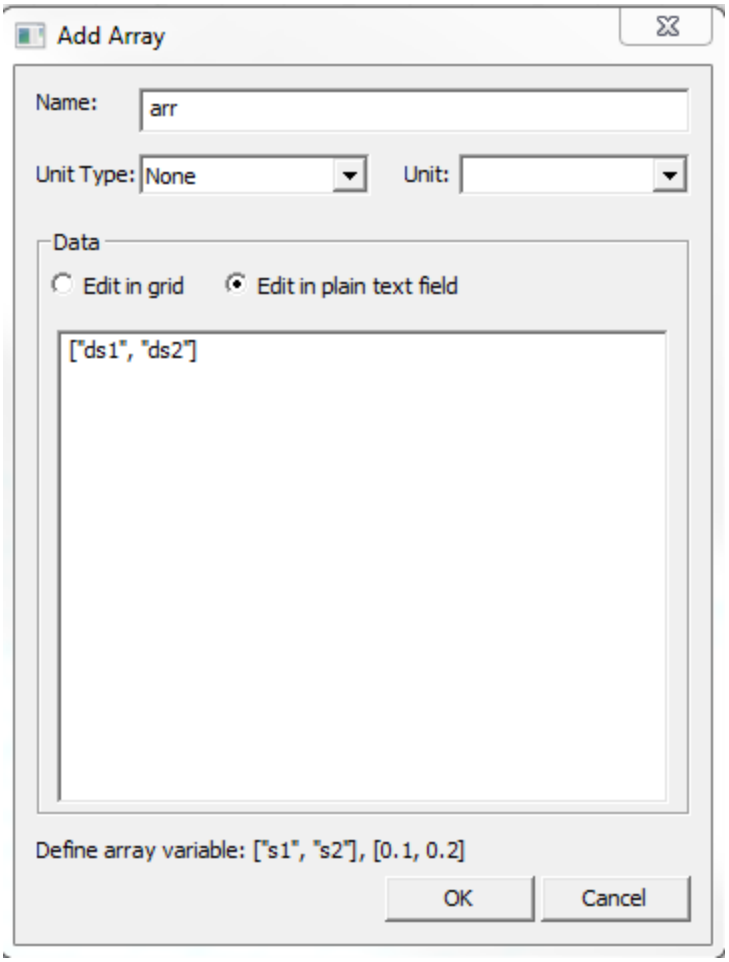
6. Specify a value and click OK.

   This displays a list of indexed data rows in the **Add Array** dialog box. You can type any data value in the cells. If you enter alphanumeric text in a cell it must be delimited by double quotes. You can edit the rows relative a row selection by clicking buttons to Add Row Above, Add Row Below, or Remove Row. All cells must contain a value.

7. When you have completed the array, click **OK** to close the dialog,

   The Array variable is listed in the **Design Properties** dialog as a Local Variable. The array variable value field includes the array contents in brackets with the unindexed data values delimited by commas.

   If you elected to edit the array Edit in plain text field in the **Add Array** dialog, the bracketed and comma delimited format is used.



Defining Mathematical Functions in RMxprt

A mathematical function is an expression that references another defined variable. A function's definition can include both expressions and variables.

The following mathematical functions may be used to define expressions:

Basic functions	**/**, **+**, **-**, *, mod (modulus), **** (exponentiation), **-** (Unary minus), **==** (equals), **!** (not), **!=** (not equals), **>** (greater than), **<** (less than), **>=** (greater than equals), **<=** (less than equals), **&&** (logical and), **||** (logical or)
Intrinsic functions	**if**, **sqn**, **abs**, **exp**, **pow**, **ln** (natural log), **sqrt**
Trigonometric expressions	**sin**, **cos**, **tan**, **asin**, **acos**, **atan**, **sinh**, **cosh**, **tanh**

The predefined variables X, Y, Z, Phi, Theta, R, and Rho must be entered as such. X, Y, and Z are the rectangular coordinates. Phi, Theta, and Rho are the spherical coordinates. R is the cylindrical radius, and Rho is the spherical radius.

If you do not specify units, all trigonometric expressions expect their arguments to be in radians, and the inverse trigonometric functions' return values are in radians. If you want to use degrees, you must supply the unit name **deg**. When the argument to a trigonometric expression is a variable, the units are assumed to be radians. These function names are reserved and may not be used as variable names.

Defining an Expression in RMxprt

Expressions are mathematical descriptions that typically contain intrinsic functions, such as sin(x), and arithmetic operators, such as +, -, *, and /, as well as defined variables. For example, you could define: x_size = 1mm, y_size = x_size + sin(x_size).

The symbol, pi (π), is the only available predefined constant. It may not be reassigned a new value.

Numerical values may be entered in Ansys' shorthand for scientific notation. For example, $5x10^7$ could be entered as **5e7**.

Using Valid Operators for Expressions in RMxprt

The operators that can be used to define an expression or function have a sequence in which they are performed. The following list shows both the valid operators and the sequence in which they are accepted (listed in decreasing precedence):

( )	parenthesis	1
!	not	2
^ (or **)	exponentiation (If you use "**" for exponentiation, as in previous software versions, it is automatically changed to "^".)	3
-	unary minus	4
*****	multiplication	5
/	division	5
+	addition	6
-	subtraction	6
==	equals	7
!=	not equals	7
>	greater than	7

<	less than	7
>=	greater than or equal to	7
<=	less than or equal to	7
&&	logic and	8
||	logic or	8

Using Intrinsic Functions in Expressions in RMxprt

RMxprt recognizes a set of intrinsic trigonometric and mathematical functions that can be used to define expressions. Intrinsic function names are reserved and may not be used as variable names.

The following intrinsic functions may be used to define expressions:

Function	Description	Syntax
abs	Absolute value (|x|)	**abs(x)**
sin	Sine	**sin(x)**
cos	Cosine	**cos(x)**
tan	Tangent	**tan(x)**
asin	Arcsine	**asin(x)**
acos	Arccosine	**acos(x)**
atan	Arc tangent. Takes a tangent value as an argument. Because there are two angles in a circle that can have the same tangent value, and atan can return only one value, it returns a value in the range between -90 degrees and +90 degrees (or between -pi/2 and pi/2 in radians).	**atan(x)**
atan2	A two-argument version of the **atan** function. Takes the y and x coordinates (including sign information) of a point as arguments and returns the angle from the X-axis. Can return angle values for the full circle (-180 degrees to +180 degrees or -pi to +pi in radians).	**atan2(y,x)**
asinh	Hyperbolic Arcsine	asinh(x)
atanh	Hyperbolic Arctangent	atanh(x)
sinh	Hyperbolic Sine	**sinh(x)**
cosh	Hyperbolic Cosine	**cosh(x)**
tanh	Hyperbolic Tangent	**tanh(x)**
even	Returns 1 if integer part of the number is even; returns 0 otherwise.	even(x)
odd	Returns 1 if integer part of the number is odd; returns 0 otherwise.	odd(x)
sgn	Sign extraction	sgn(x)
exp	Exponential ($\mathbf{e}^{x}$)	**exp(x)**

pow	Raise to power (x^y)	pow(x,y)
if	If	**if(cond_exp,true_exp, false_exp)**
pwl	Piecewise Linear with linear extrapolation on x. (pwl can be used with datasets for Design Variables but not for Project variables).	pwl **(dataset_exp, variable)**
pwlx	Piecewise Linear x with linear extrapolation on x	pwlx **(dataset_exp, variable)**
pwl_periodic	Piecewise Linear with periodic extrapolation on x	pwl_periodic **(dataset_exp, variable)**
sqrt	Square Root	**sqrt(x)**
ln	Natural Logarithm (The "log" function has been discontinued. If you use "log(x)" in an expression, the software automatically changes it to "ln(x)".)	**ln(x)**
log10	Logarithm base 10	log10(x)
int	Truncated integer function	int(x)
nint	Nearest integer	nint(x)
max	Maximum value of two parameters	max(x,y)
min	Minimum value of two parameters	min(x,y)
mod	Modulus	mod(x,y)
rem	Returns the fractional part of a decimal number such that rem(x) = x-int(x)	rem(x)
clp	Closest point interpolation. Note: If used with a large 3D dataset, clp function will degrade.	clp (*datasetName*, X,Y,Z)

Note	If you do not specify units, all trigonometric functions interpret their arguments as radians. Likewise, inverse trigonometric functions' return values are in given in radians. When the argument to a trigonometric expression is a variable, the units are assumed to be radians. If you want values interpreted in degrees, supply the argument with the unit name **deg**

Using Piecewise Linear Functions in Expressions in RMxprt

The following piecewise linear intrinsic functions are accepted in expressions:

```
pwl (dataset_expression, variable)
pwlx (dataset_expression, variable)
pwl_periodic (dataset_expression, variable)
```

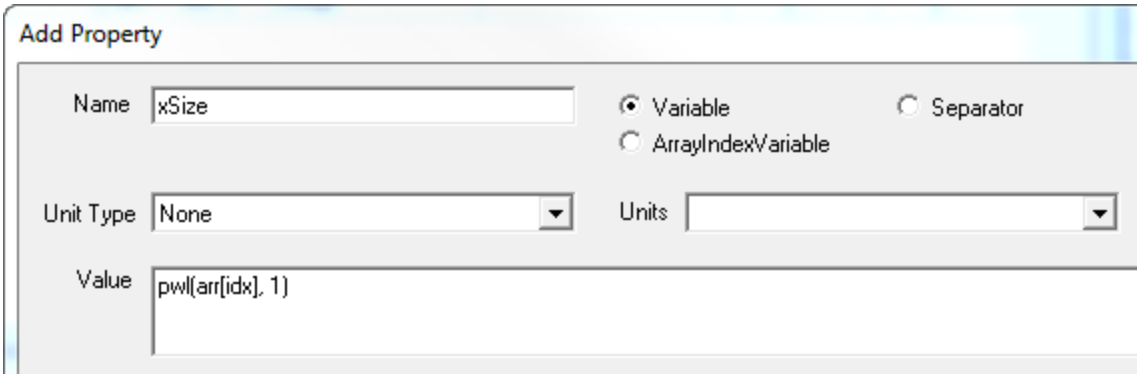
The **pwl** and **pwlx** functions interpolate along the x-axis and returns a corresponding y value. The **pwl_periodic** function also interpolates along the x-axis but periodically.

You can use **pwl** in an expression that uses array variables and datasets for uses such as a frequency dependent material property. For example, you specify BulkConductivity as:

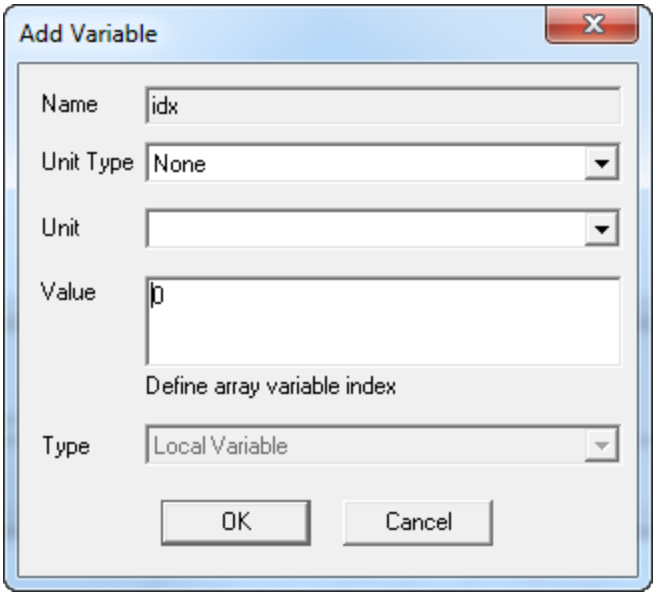
`pwlx($dsArr[$dsIndex], Freq)` where `$dsArr=["$ds1", "$ds2"]`

You can create a Design Variable representing dimension `xSize` as `pwl(arr[idx], 1)` where "`arr`" is an array variable and `idx` is an array index variable.

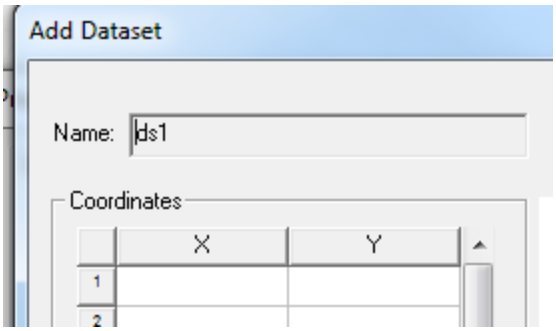
In this case, creating a variable named xSize with pwl(arr[idx], 1) like this:



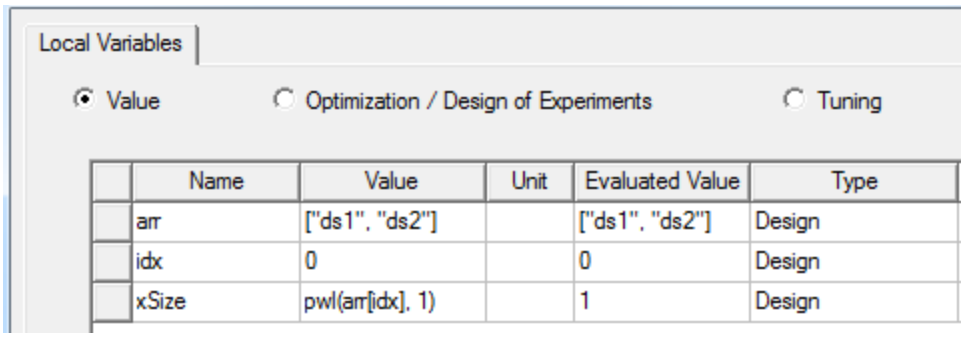
This value for xSize automatically opens a dialog first for the idx index variable:



And then dialogs automatically open for each dataset variable implicit for the predefined array variable:



Then the **Design Properties** dialog shows as follows:



Note: While you can use pwl for a design variable dataset, it does not work for a project variable dataset.

Using Dataset Expressions in RMxprt

Dataset expressions take the following form:

$$dset((x_0,y_0), \ldots, (x_n,y_n))$$

These expressions may be used as the first parameter to piecewise linear (**pwl**, **pwlx** and **pwl_periodic**) functions, and may also be assigned to variables, in which case the variable may be used as the second parameter to **pwl**, **pwlx** and **pwl_periodic** functions.

Dataset expressions are derived from a series of points in a plot created in the **Datasets** dialog box. (Refer to Adding Datasets for related information on working with datasets.) Each plot consists of straight line segments whose vertices represent their end points. A curve is fitted to the segments of the plot, and an expression is derived from the curve that best fits the segmented plot. The created expression is then used in the piecewise linear intrinsic functions.

Assigning Variables in RMxprt

To assign a variable to a parameter in RMxprt:

- Type the variable name or mathematical expression in place of a parameter value in a **Value** box.

  If you typed a variable name that has not been defined, the **Add Variable** dialog box appears, enabling you to define the design variable.

  If you typed a variable name that included the **$** prefix, but that has not been defined, the **Add Variable** dialog box appears, enabling you to define the project variable.

  You can assign a variable to nearly any design parameter assigned a numeric value in RMxprt.

Choosing a Variable to Optimize in RMxprt

Before a variable can be optimized, you must specify that you intend for it to be used during an optimization analysis in the **Properties** dialog box.

1. If the variable is a design variable, click **RMxprt>Design Properties**. You can also access the Design variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Design Variables**.

   If the variable is a project variable, click **Project>Project Variables**. You can also access the Project variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**.

   The **Properties** dialog box appears.

2. Click the tab that lists the variable you want to optimize.
3. Click the row containing the variable you want to optimize.

Note	Dependent variables cannot be optimized.

4. Select the **Optimization/Design of Experiments** option.
5. For the variable you want to optimize, select **Include**.

   The selected variable is now available for optimization in an optimetrics setup defined in the current design or project.
6. Optionally, override the default minimum and maximum values that Optimetrics will use for the variable in every optimization analysis. During optimization, the optimizer does not consider variable values that lie outside of this range.
7. Click **OK**.

Related Topics

Setting up an Optimization Analysis

Including a Variable in a Sensitivity Analysis in RMxprt

Before a variable can be included in a sensitivity analysis, you must specify that you intend for it to be used during a sensitivity analysis in the **Properties** dialog box.

1. If the variable is a design variable, click **RMxprt>Design Properties**. You can also access the Design variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Design Variables**.

   If the variable is a project variable, click **Project>Project Variables**. You can also access the Project variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**.

   The **Properties** dialog box appears.
2. Click the tab that lists the variable you want to include in the sensitivity analysis.
3. Click the row containing the variable you want to include in the sensitivity analysis.

Note	Dependent variables cannot be included in a sensitivity analysis.

4. Select the **Sensitivity** option.
5. For the variable you want to include in the sensitivity analysis, select **Include**.

   The selected variable is now available for sensitivity analysis in a sensitivity setup defined in the current design or project.

6. Optionally, override the default minimum and maximum values that Optimetrics will use for the variable in every sensitivity analysis. During sensitivity analysis, Optimetrics will not consider variable values that lie outside of this range.
7. Optionally, override the default initial displacement value that Optimetrics will use for the variable in every sensitivity analysis. During sensitivity analysis, Optimetrics will not consider a variable value for the first design variation that is greater than this step size away from the starting variable value.
8. Click **OK**.

Related Topics

Setting up a Sensitivity Analysis

Choosing a Variable to Tune in RMxprt

Before a variable can be tuned, you must specify that you intend for it to be tuned in the **Properties** dialog box.

1. If the variable is a design variable, click **RMxprt>Design Properties**.

   If the variable is a project variable, click **Project**>**Project Variables**.

   The **Properties** dialog box appears.
2. Click the tab that lists the variable you want to tune.
3. Click the row containing the variable you want to tune.

Note	Dependent variables cannot be tuned.
4. Select the **Tuning** option.
5. For the variable you want to tune, select **Include**.
6. Click **OK**.

   The selected variable is now available for tuning in the **Tune** dialog box.

Related Topics

Tuning a Variable

Including a Variable in a Statistical Analysis in RMxprt

Before a variable can be included in a statistical analysis, you must specify that you intend for it to be used during a statistical analysis in the **Properties** dialog box.

1. If the variable is a design variable, click **RMxprt>Design Properties**. You can also access the Design variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and

Design Xplorer Setup. Click **Edit Variables** and from the menu select **Edit Design Variables**.

If the variable is a project variable, click **Project**>**Project Variables**. You can also access the Project variables from a menu in the lower left corner of the following Optimization dialogs: **Parametric**, **Optimization**, **Sensitivity**, **Statistical**, **Design of Experiments**, and **Design Xplorer Setup**. Click **Edit Variables** and from the menu select **Edit Project Variables**.

The **Properties** dialog box appears.

2. Click the tab that lists the variable you want to include in the statistical analysis.
3. Click the row containing the variable you want to include in the statistical analysis.

Note	Dependent variables cannot be included in a statistical analysis.

4. Select the **Statistics** option.
5. For the variable you want to include in the statistical analysis, select **Include**.

   The selected variable is now available for statistical analysis in a statistical setup defined in the current design or project.
6. Optionally, override the distribution criteria that Optimetrics will use for the variable in every statistical analysis.
7. Click **OK**.

Related Topics

Setting up a Statistical Analysis

Working with Materials in RMxprt

RMxprt shares many common functions related to materials and material handling with **Maxwell3D** and **Maxwell2D**. For general topics such as, Assigning Materials, Searching for Materials, Adding New Materials, and Viewing and Modifying Material Attributes, see the topics in Maxwell help under **Assigning Materials**.

RMxprt includes a material library containing common materials used in motor design. However, this library needs to be configured so that it is automatically loaded for any new **RMxprt** design. Once the library is configured, you can add, remove, and edit RMxprt materials in two main ways:

- Using the **Tools>Edit Libraries>Materials** menu command.
- Right-clicking **Materials** in the project tree and selecting **Edit Library**.

Editing definitions from the project window does not modify the configured libraries for any particular design. To consider the current design, use the **Tools>Edit Libraries** option. Doing so ensures that new libraries are added to the configured list for the current design. If you edit materials from this command for the current and then export them, they will also be available to assign to objects in other designs.

Related Topics

Material Library Management for RMxprt

Material Library Management for RMxprt

The three most crucial electromagnetic materials in the electric machine are soft-magnetic material (silicon steel sheet), hard-magnetic material (permanent magnet) and electromagnetic wire. It is convenient to create a data file library for them for quick selection while inputting design data.

Soft-Magnetic Materials

The stator and the rotor iron cores in the electric machine are generally laminated with punched sheets of nonlinear soft-magnetic silicon steel. Some special types of electric machines, such as moment motor, turbo-generator etc., use integrated solid rotor iron core of soft-magnetic material. For magnetic field analysis and core loss analysis of the electric machine, the magnetization characteristics (**B-H Curve**) and the loss characteristics (**B-P Curve**) of the iron-core material must be defined. The dialog boxes to do so are accessed from the **View/Edit Material** window, which, in turn is accessed from the **Edit Libraries** window. Access to the window for editing the B-H curve is enabled when you set the Magnetic Permeability value to nonlinear (rather than simple or Anisotropic). When you set the value to nonlinear, the value field changes to a B-H Curve button. Click the B-H curve button to open the window.

For the loss characteristics (B-P Curve), you first set the **Core Loss Model** of the material to **Electrical Steel** (rather than **None** or **Power Ferrite**) as a material property in the **View/Edit Material** window. When you do so, this enables the **Calculate Properties** selection for drop down

at the bottom of the window. Select **Core Loss Coefficient** from the drop down menu to open the B-P Curve window.

Related Topics

Adding New Materials to an RMxprt Project
Setting the Material Threshold for RMxprt
Assigning Materials
Removing Materials
Validating Materials
Sorting Materials
Viewing and Modifying Material Attributes
Copying Materials
Exporting Materials to a Library
Calculating Properties for Core Loss in RMxprt (BP Curve)

Adding New Materials to an RMxprt Project

You can add a new material to a project or to the global user-defined material library. To make the new project material available to all projects, you must export the material to a global user-defined material library.

To assign a material to an object:

1. Click **Tools>Edit Libraries>Materials**.
   - In the project tree, you can also right-click **Materials**, and select **Edit Library**.

   The **Edit Libraries** dialog box appears.
1. Click **Add Material**.

   The **View/Edit Material** dialog box appears.
2. Type a name for the new material in the **Material Name** text box.
3. In the **View/Edit Material for** section, select whether this material should be available for the **Active Design** only, for **This Product** only or for **All Products**. The selection makes a difference in the properties displayed.

Note	When you select certain **Type** or **Value** options, additional parameters appear beneath some properties in the same way that **Magnitude** appears beneath **Magnetic Coercivity**. As necessary, specify values for any additional parameters that appear.

4. In the **Properties of the Material** table, specify the **Type** and the **Value** for the following material properties, displayed when **Active Design** is selected:
   - Relative Permeability.
   - Bulk Conductivity
   - Magnetic Coercivity (including the Magnitude of the vector)

- Core Loss Model - model selections for this property may enable access to coefficient calculation windows, and enable additional properties.
- Mass Density

If you select **This Product**, additional fields are displayed.

- Relative Permittivity
- Dielectric Loss Tangent
- Magnetic Loss Tangent
- Composition

5. Optionally, change the **Units** for any of the properties.
6. Click **Validate Material** to verify the settings you have specified are valid for the existing properties.
   If the material setup is valid, a green check mark appears below the **Validate Material** button.
7. Click **OK** to save the new material.
   The **Edit Libraries** dialog box reappears, with the new material added to the list of materials.
8. Click **OK** to close the **Edit Libraries** dialog box.

Relative Permittivity for an RMxprt Material

Specify the following for Relative Permittivity and specify the units.

Type	Value
Simple	Type a value for the **Relative Permittivity**.
Anisotropic	The following two parameters appear: • T(1,1) • T(2,2) Type a simple value for each.
Nonlinear	The material's relative permittivity varies according to its associated D-E curve. **Note** Not valid in RMxprt.

Note	This property is not used in RMxprt design analysis, but it will be transferred to Maxwell 3D Design automatically when the design is created by RMxprt.

Relative Permeability for a Maxwell or RMxprt Material

Select one the following for relative permeability and specify the units:

Type	Value
Simple	Type a value for the **Relative Permeability**.
Anisotropic	The following parameters appear: • T(1,1) • T(2,2) • T(3,3) Select either a **Simple** or **Nonlinear** Type for each of these parameters.
Nonlinear	Click **BH Curve**, and specify the coordinates for the BH-curve (or curves if multiple temperature dependencies are used) in the BH Curve dialog box. Also enables X, Y, and Z Component unit vector fields for Magnetic Coercivity.

Note	The Anisotropic type is not used in RMxprt design.

Specifying BH Curves for Nonlinear Relative Permeability

For Maxwell 2D and 3D eddy current, magnetostatic and transient solution types, when you define a new material or edit an existing material with a nonlinear relative permeability in the **View/Edit Materials** window, you need to specify the magnetization characteristics using BH curves.

You can also model temperature dependencies of the nonlinear material property by providing two or more temperature-dependent BH curves. For temperature dependencies of permanent magnetic materials, refer to *Interpolation of Temperature-Dependent Demagnetization Curves*.

Note	The thermal modifier for permeability is disabled when temperature-dependent BH curves are specified.

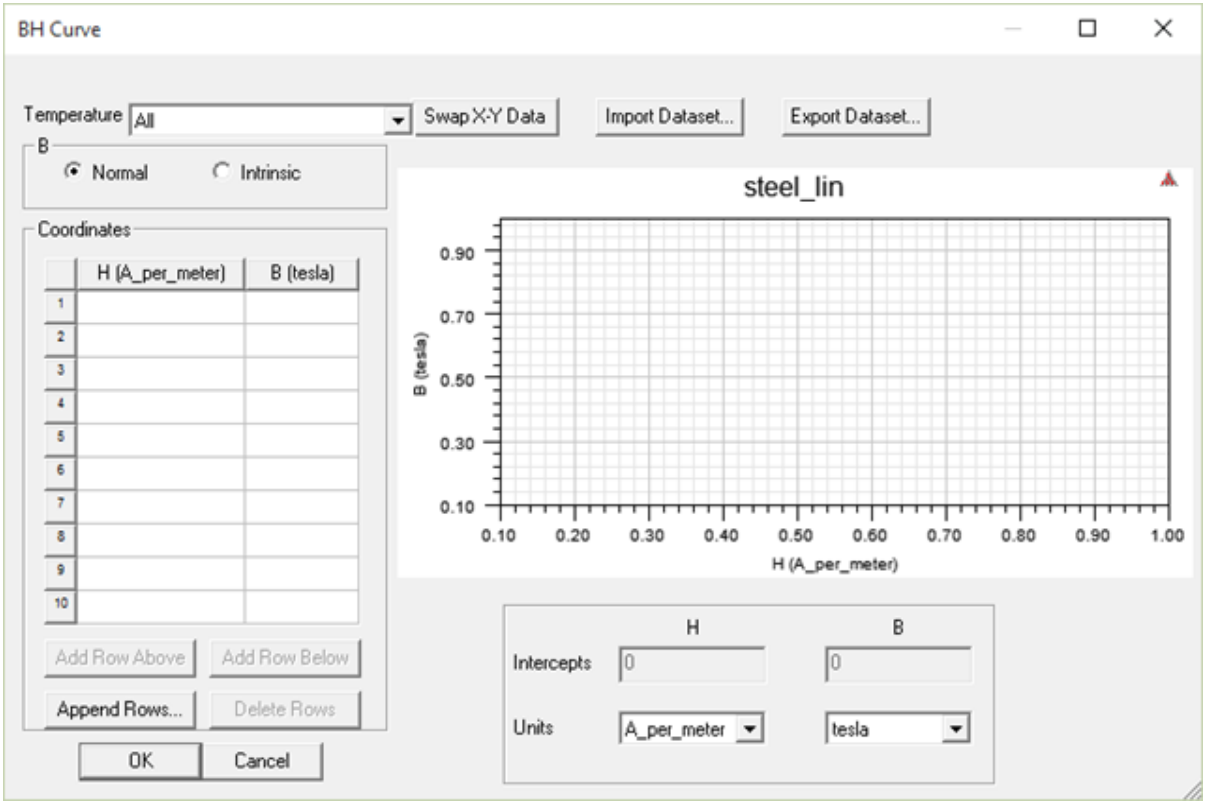
If a user-input BH curve is used without smoothing, the non-smoothed BH curve is treated as piece-wise linear between user-input data points inside the software, which may cause nonlinear convergence issues. However, if you chose to use a smoothed BH curve constructed by the software, the operating point may not exactly reside on the user-input BH curve. **Smooth BH curve** is not selected by default. You can select it from the **Solve Setup**, on the **Solver** panel. For isotropic nonlinear materials, you can force Maxwell to use the original user-input data points without smoothing by software by clearing the **Smooth BH Curve** option.

1. Open the **View / Edit Materials** dialog from the **Edit Materials** window either by:
   - Selecting an existing material that you need to edit, and click **View / Edit Material**.
   - Clicking **Add Material.**

   Either of these actions open the **View/ Edit Materials** dialog box.
2. For the **Relative Permeability** property do one of the following (depending on the type of material you are defining):
   - Select **Nonlinear** as the **Type**. A **B-H Curve** button appears in the **Value** column.
   - Select **Anisotropic** as the **Type** to display the additional parameters: **T(1,1)**, **T(2,2)**,

T(3,3). Selecting **Nonlinear** for any of these additional parameters also causes a **B-H Curve** button to appear in the **Value** columns.

3. You can Input BH curve(s) by clicking a **B-H Curve** button to open the **BH Curve** dialog box.



4. Set the **Units** for **H** and **B** by selecting from the drop down menus. If temperature-dependent BH curves are used, the selection of units applied to all of the temperature-dependent curves
5. Choose the type of curve you want to define by selecting either **Normal** or **Intrinsic**.

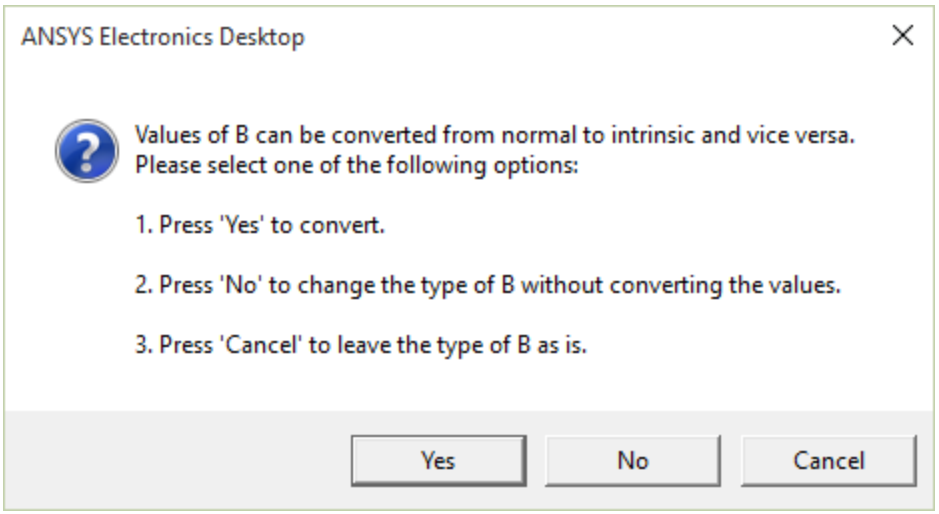
Note	Selection between **Normal** or **Intrinsic** BH curve is not supported when temperature-dependent BH curves are used. Temperature-dependent curves are restricted to the type **Normal**.

For a material property without an existing BH curve definition, the dialog opens with an empty list of coordinates and the default type will be **Normal**. For a property with existing BH curve definition, the selected radio button corresponds to the existing B type.

Note	• The **Intrinsic** BH curve is supported only in Maxwell 2D/3D magnetostatic, eddy current, and transient design types. A material property defined using an Intrinsic BH curve will fail validation check in all the other product/design types. • When an **Intrinsic** BH curve is added, the **Relative Permeability Value**

	button label in the **View/Edit Material** dialog box changes to **Bi-H Curve** as visual indication of the type of curve currently defined for the material.

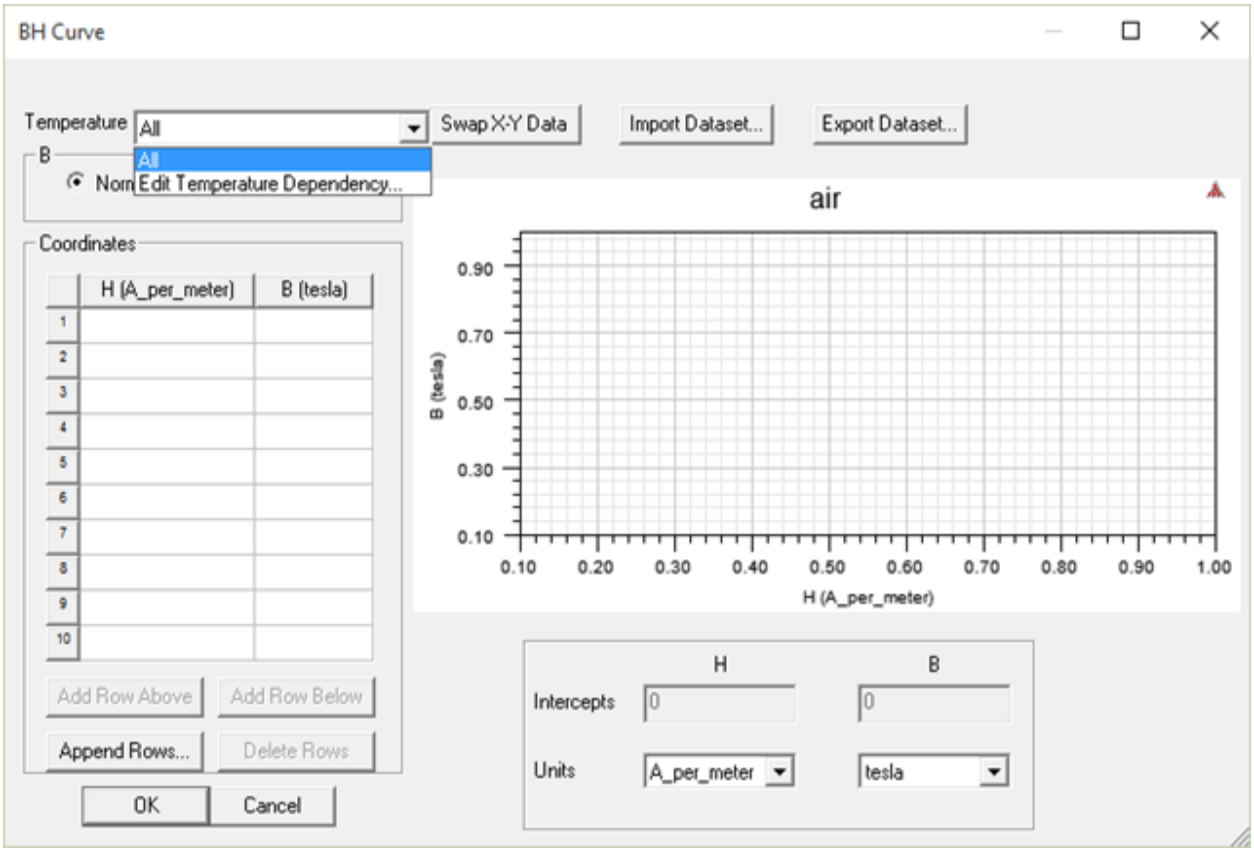
You can change the type at any time. For an existing curve, validation checks are performed on the coordinate list when you attempt to change the type. If the data is not valid, an error message will be displayed and the type of B will not be changed. If data is valid, a query dialog box displays asking if the coordinates should be converted.



Pressing **No** can be used, fer example, when users have specified the BH coordinates and then realize they haven't select the desired type.

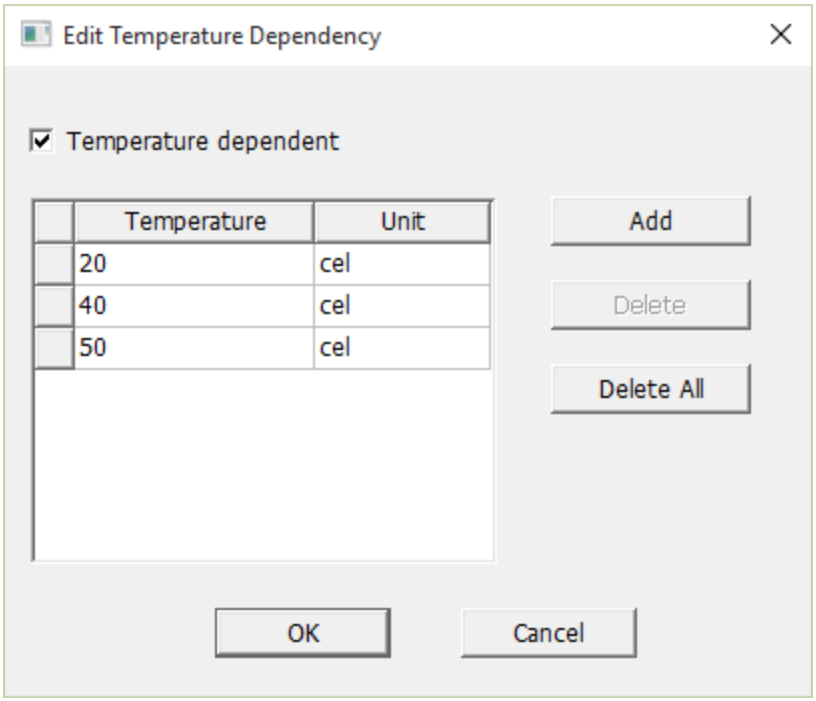
Note	Changing the type of the BH curve invalidates all solution data

6. The initial selections in the **Temperature** drop-down menu include **All** and **Edit Temperature Dependency…**.



The default selection is **All**, which means the BH curve applies to all temperatures – in other words, the BH curve is *not* temperature dependent. For this case, continue with step 8.

7. Optionally, for Maxwell 2D and 3D eddy current, magnetostatic and transient solution types, you can model temperature dependencies of the nonlinear material property by providing multiple temperature-dependent BH curves. At least two BH curves must be defined when using this feature. If you want to change the **Temperature** setting to specify temperature-dependent BH curves:
    a. Select **Edit Temperature Dependency** to open the Edit Temperature Dependency dialog box.



b. Use the **Add** button to add rows to the table. In each row, enter the desired **Temperature**, and select the appropriate **Unit**. The **Delete** button can be used to remove selected rows from the table. **Delete All** removes all of the rows.

c. Check **Temperature dependent** to enable temperature-dependent BH curves to be added.

d. Click OK to return to the **BH Curve** dialog box.

The **Temperature** drop-down now lists all of the temperatures you specified above. Select the one for which you want to enter B and H coordinate values in the following step. Also, a **Show all curves** check box is now present. Checking this enables all of the BH curves to be displayed simultaneously.

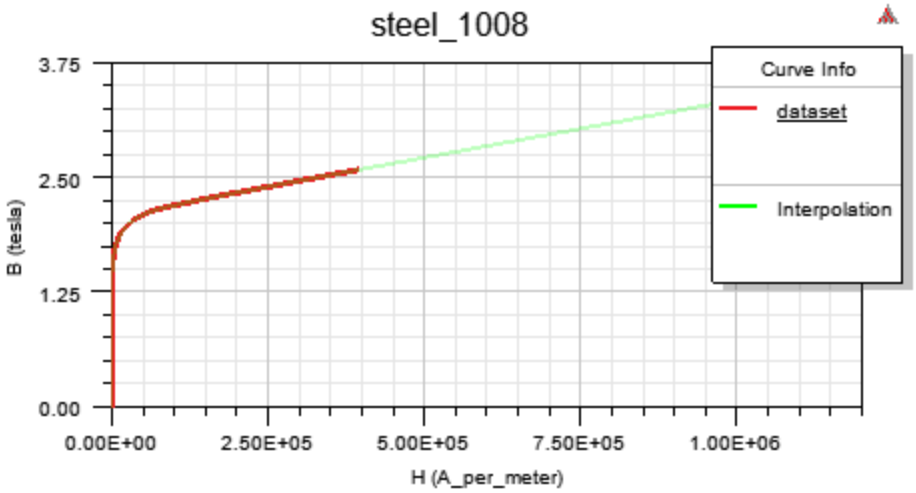
e. Select the temperature for which you want to add (or edit) BH curve values, then continue with the next step.

8. Enter B and H values in each row of the **Coordinates** table. Placing the cursor in a table cell enables the **Add Row Above**, **Add Row Below**, and **Delete Rows** buttons. **Append Rows** allows you to specify the number of rows to append to the table.

Note	• When using temperature-dependent BH curves, the settings in the BH Curve dialog apply to the currently selected temperature value. • When adding (or editing) temperature-dependent BH curves, repeat this step for each specified temperature as needed.

Note the following requirements for creating a valid curve:

- For a **Normal** BH curve, the slope of the curve can not be less than that of free space anywhere along the curve.
- For an **Intrinsic** BH curve, the slope of the curve can not be less than 0.
- The value of B must increase along the curve.
- The initial value of B must be 0 (zero).
- Since BH operating points in the FEA solution may extend beyond the input BH data set, the BH data set is extrapolated in Maxwell. The slope of the last two user-defined data points is used to extrapolate the BH curve, and thus should be as close to μ_0 as possible.
- The data points representing the BH curve should have enough points for accurate representation of the curve. Twenty (20) or more points should be specified with increased representation on the "knee" of the curve.
- Normal BH curves with a positive B value at the first point will be extrapolated (interpolation). Intrinsic curves are not extrapolated.



As you enter values, the graph is updated. If you are using multiple temperature-dependent BH curves, a **Show all curves** check box is present. Checking this box enables all of the temperature-dependent BH curves to be displayed simultaneously.

- Optionally, click **Import Dataset** to import BH curve data from a file, and if they are in the wrong columns, click **Swap X-Y Data** to switch the B values and H values in the graphics display. You can also use the SheetScan tool to extract curve data from sources such as manufacturer datasheets to a dataset, which can then be exported to a tab-delimited file, and imported via **Import Dataset**. (Refer to *Adding Datasets* and *Exporting Datasets* for related information on working with datasets. Refer to *Using SheetScan* for working with the SheetScan tool.)

Note	When using temperature-dependent BH curves, selecting **Import Dataset** will import a single curve for the currently selected temperature.

9. When finished entering data, click **OK** to close the window.

   An error message displays if a slope is out of tolerance, identifying the data points between which the slope is less than that of free space. Out of tolerance data points must be corrected before you can successfully exit the dialog box. If the slope of the last two points is more than twice the permeability of free space, a warning message is issued during the solution. The slope of the last 2 points should be as close to μ_0 as possible to represent a fully saturated material.

The BH curve (or curves if multiple temperature dependencies are used) you have defined is associated with the Relative Permeability property of the material.

Note	When an **Intrinsic** BH curve is added, the **Relative Permeability Value** button label changes to **Bi-H Curve** as visual indication of the type of curve currently defined for the material.

Related Topics

Setting a Thermal Modifier

Assigning Material Property Types

Temperature-Dependent BH Curves

Temperature-Dependent BH Curves

While a material's temperature dependency can be specified via the thermal modifier, which is described in terms of a mathematical expression, this might not be good enough for applications where it is difficult to describe the temperature dependency in terms of an expression. For such cases, an alternative method is available to let users directly input data for multiple, temperature-dependent, BH curves. Maxwell uses this data to derive a corresponding BH curve for each specified temperature using an interpolation algorithm as follows:

Suppose there are m temperature dependent curve inputs in the form of:

$$T_0 :: \quad (0,0), (h_{01},\ b_{01}) \cdots\cdots (h_{0i},\ b_{0i}) \cdots\cdots (h_{0N_0},\ b_{0N_0})$$

$$\cdots\cdots\cdots\cdots\cdots$$

T_k :: $(0,0), (h_{k1},\ b_{k1}) \cdots\cdots (h_{ki},\ b_{ki}) \cdots\cdots (h_{kN_k},\ b_{0N_k})$

$\cdots\cdots\cdots\cdots\cdots$

T_m : : $(0,0), (h_{m1},\ b_{m1}) \cdots\cdots (h_{mi},\ b_{mi}) \cdots\cdots (h_{mN_m},\ b_{mN_m})$

Where T_0 , T_k *and* T_m ... $(T_0 < T_k < T_m)$ are reference temperatures at which BH curves were measured.

The number of *(b,h)* pairs in each BH curve can be different.

In Maxwell, if the local temperature of temperature-dependent nonlinear magnetic material is *T*.. $(T_k < T < T_{k+1})$, the local BH curve can be obtained by interpolation:

If $T \le T_0$, BH curve input at temperature T_0 is applied.

If $T \ge T_m$, BH curve input at temperature T_m is applied.

If $T_k < T < T_{k+1}$, $(h_i,\ b_i)$ pairs are calculated as:

For a given h_i , $b_{k,i}$ *and* $b_{k+1,\ i}$ are magnetic flux densities obtained from inputted BH curves at temperatures T_k *and* T_{k+1} , and b_i is obtained by interpolation:

$$b_i = b_{k,i} + \frac{T - T_k}{T_{k+1} - T_k}(b_{k+1,i} - b_{k,i})$$

Note	Thermal modifier for permeability is disabled when temperature-dependent BH curves are specified.

Related Topics

Setting a Thermal Modifier

Adding Datasets

View/ Edit materials dialog

Set Ambient Temperature

Interpolation of Temperature-Dependent Demagnetization Curves

For an intrinsic demagnetization curve at reference temperature T_0 expressed by a data set (h_i, b_i), the demagnetization curve at temperature *T* can be derived by scaling *h* and *b* independently as (h_i•*Q*(*T*), b_i•*P*(*T*)), where temperature dependent scaling factors *P*(*T*) and *Q*(*T*) can be specified via the thermal modifier. This algorithm, called shape preserving algorithm, can be found in *Temperature Dependent Nonlinear Permanent Magnets.* However, this mathematical expression might not be good enough for applications where it is difficult to describe the temperature dependency in terms of an expression. In addition, the measured multiple temperature-dependent intrinsic demagnetization curves might not be shape-preserved in some materials. For such cases, an alternative method is available to let users directly input data for multiple temperature-

dependent demagnetization curves. Maxwell uses this data to derive a corresponding demagnetization curve for a specified temperature using an interpolation algorithm as follows.

Suppose there are *m* inputs of temperature-dependent intrinsic demagnetization curve in the form of:

$$T_0 :: (h_{ci\_0},\ 0), (h_{01},\ b_{01}) \cdots\cdots (h_{0i},\ b_{0i}) \cdots\cdots (0,\ b_{0N_0})$$

$$\cdots\cdots\cdots\cdots\cdots$$

$$T_k :: (h_{ci\_k},\ 0), (h_{k1},\ b_{k1}) \cdots\cdots (h_{ki},\ b_{ki}) \cdots\cdots (0,\ b_{kN_k})$$

$$\cdots\cdots\cdots\cdots\cdots$$

$$T_m :: (h_{ci\_m},\ 0), (h_{m1},\ b_{m1}) \cdots\cdots (h_{mi},\ b_{mi}) \cdots\cdots (0,\ b_{mN_m})$$

Where T_0 , T_k *and* T_m ... $(T_0 < T_k < T_m)$ are reference temperatures at which demagnetization curves were measured. The flux density of the end point b_{kN_k} is the residual b_{r_k} for curve *k*.

The number of (h_{ki}, b_{ki}) points, N_k +1, in each demagnetization curve can be different.

In Maxwell, if the local temperature of temperature-dependent nonlinear magnetic material is *T*.. $(T_k < T < T_{k+1})$, the local demagnetization curve can be obtained by interpolation:

If $T \le T_0$, the demagnetization curve at temperature T_0 is returned as the interpolated demagnetization curve.

If $T \ge T_m$, the demagnetization curve at temperature T_m is returned as the interpolated demagnetization curve.

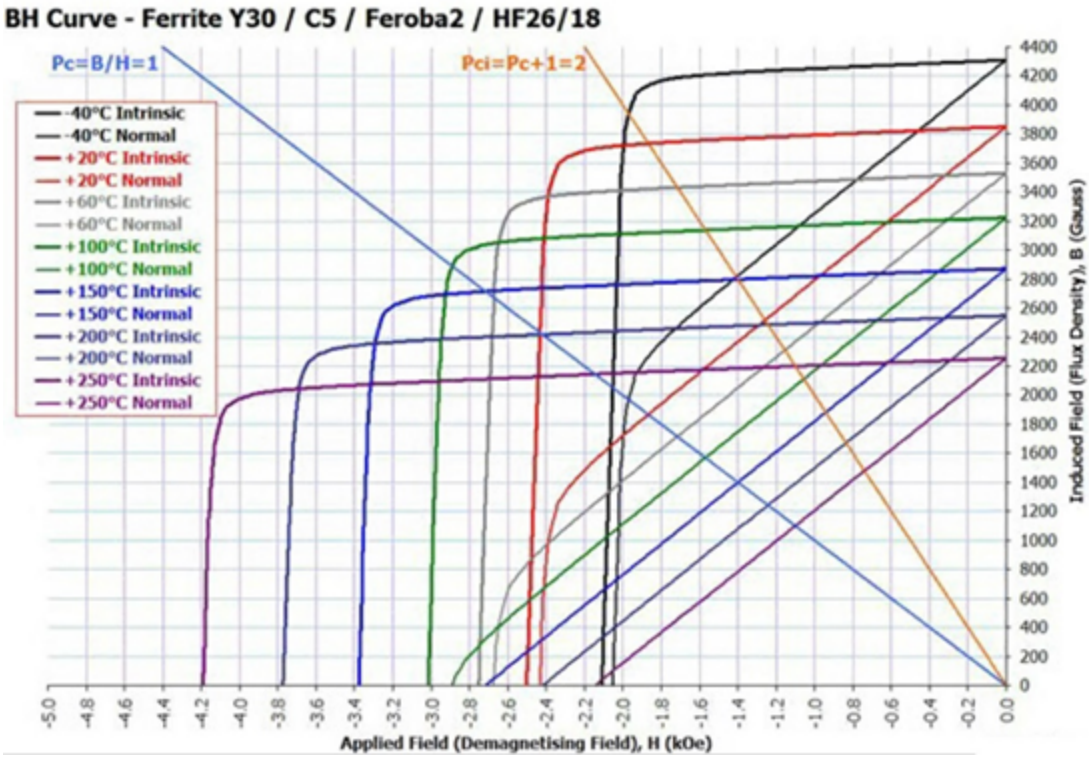
If $T_k < T < T_{k+1}$, the interpolated demagnetization curve at temperature T can be derived by following steps:

1. Get temperature interpolating coefficient: $c_T = (T - T_k) / (T_{k+1} - T_k)$.
2. Get interpolated intrinsic coercivity: $h_{ci} = (1 - c_T)\ h_{ci\_k} + c_T\ h_{ci\_k+1}$.
3. Get interpolated residual: $b_r = (1 - c_T)\ b_{r\_k} + c_T\ b_{r\_k+1}$.
4. Derive demagnetization curve at temperature *T* using shape preserving algorithm based on the curve at T_k, which is expressed as $(h_{ki} \cdot Q_0,\ b_{ki} \cdot P_0)$, where $Q_0 = h_{ci} / h_{ci\_k}$, and $P_0 = b_r / b_{r\_k}$.
5. Derive demagnetization curve at temperature *T* using shape preserving algorithm based on the curve at T_{k+1}, which is expressed as $(h_{k+1\_i} \cdot Q_1,\ b_{k+1\_i} \cdot P_1)$, where $Q_1 = h_{ci} / h_{ci\_k+1}$, and $P_1 = b_r / b_{r\_k+1}$.
6. Derive the weighted average curve from the curve expressed by $(h_{ki} \cdot Q_0,\ b_{ki} \cdot P_0)$ and the curve expressed by $(h_{k+1\_i} \cdot Q_1,\ b_{k+1\_i} \cdot P_1)$ with weighting factors of $(1 - c_T)$ and c_T, respectively.

We can input temperature-dependent demagnetization curves in both normal and intrinsic formats. When normal demagnetization curves are input, we will transfer them to intrinsic curves

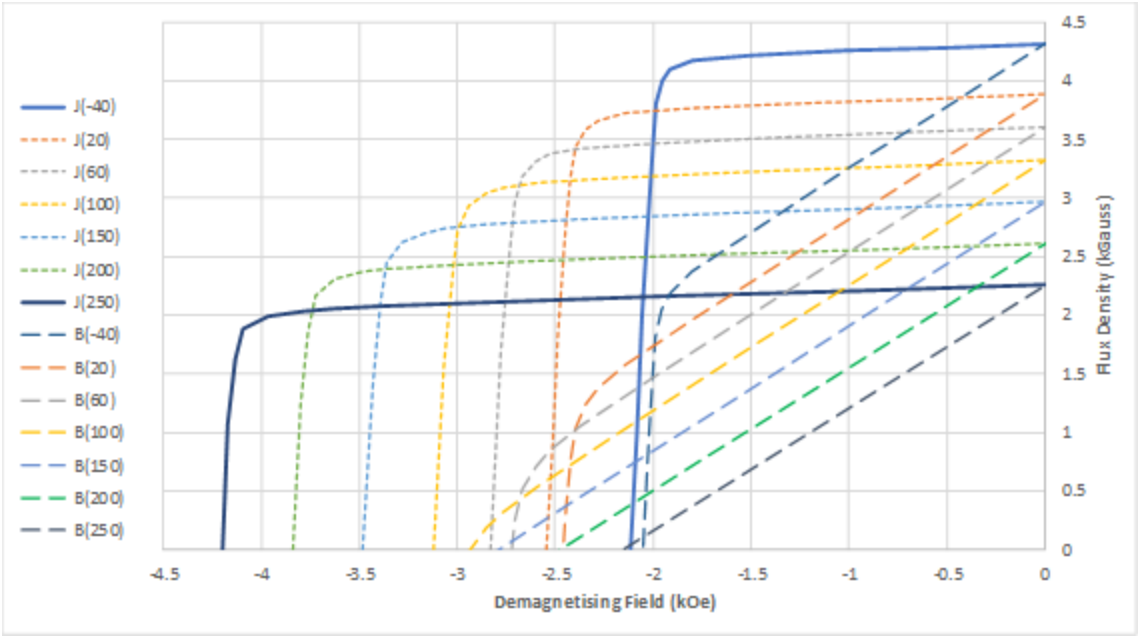
and extend them with additional estimated data sets. Since the extended curve parts are approximated, it is suggested to directly input intrinsic curves with different temperatures as basic curves for interpolation, if they are available.

For example, the material Ferrite Y30 has BH curves with Br decreasing and Hci increasing as temperature increasing, as shown below.



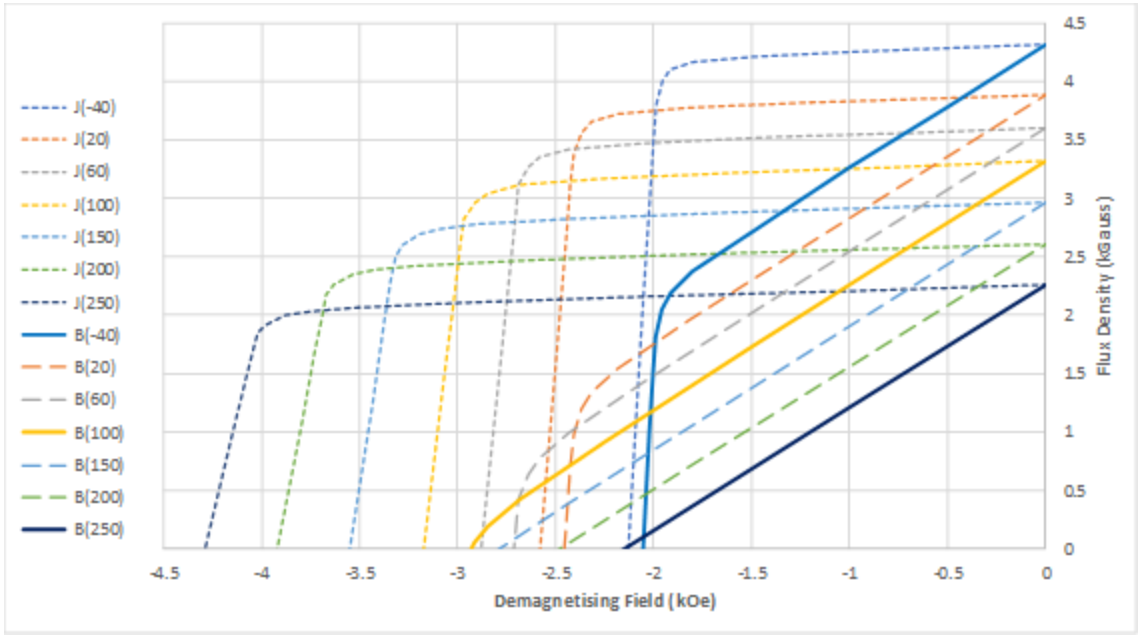
Temperature-dependent BH curves for Ferrite Y30

If we input two intrinsic curves of the minimum and maximum temperatures, we can interpolate the intrinsic curves of other temperatures, and derive normal curves, as shown below.



Input two intrinsic curves in (solid lines)
Interpolated intrinsic curves (short-dashed lines)
Derived normal curves (long-dashed lines)

If intrinsic curves are not available, we need to input three normal curves with minimum and maximum temperatures, as well as the curve with maximum Hc. In such cases, the estimated and interpolated intrinsic curves, and derived normal curves, are as shown below.



Input three normal curves (solid lines) Estimated and interpolated intrinsic curves (short-dashed lines) Derived normal curves (long-dashed lines)

Note: Thermal modifier for permeability is disabled when temperature-dependent BH curves are specified.

Related Topics

Setting a Thermal Modifier
Adding Datasets
View/ Edit materials dialog
Temperature Dependent Nonlinear Permanent Magnets

Bulk Conductivity for an RMxprt Material

Specify the following for bulk conductivity and specify the units:

Type	Value
Simple	Type a value for the **Bulk Conductivity**.
Anisotropic	The following parameters appear: • T(1,1) • T(2,2) You can enter a simple value for each of these parameters. **Note** The Anisotropic type is not used in RMxprt designs.
Nonlinear	The material's bulk conductivity varies according to its associated J-E curve. **Note** The Nonlinear type is not used in RMxprt designs.

Dielectric Loss Tangent for RMxprt Material

Specify the following for dielectric loss tangent.

Type	Value
Simple	Type a value for the**Dielectric Loss tangent**.
Anisotropic	The following two parameters appear: • T(1,1) • T(2,2) Type a simple value for each.

Note	This property is not used in RMxprt design analysis, but it will be transferred to Maxwell 3D Design automatically when the design is created by RMxprt.

Magnetic Loss Tangent for RMxprt Material

Type	Value
Simple	Type a value for the **Bulk Conductivity**.
Anisotropic	The following two parameters appear: • T(1,1) • T(2,2) Type a simple value for each.

Note	This property is not used in RMxprt design analysis, but it will be transferred to Maxwell 3D Design automatically when the design is created by RMxprt.

Magnetic Coercivity for Maxwell and RMxprt Materials

Specify the following for magnetic coercivity and specify the units:

Type	Value
Vector	Appears by default. • If the Relative Permeability Type is either **Simple** or **Anisotropic**, enter a **Value** for the **Magnitude**. • If the Relative Permeability Type is Nonlinear, Magnitude becomes uneditable, and three additional fields of Type **Unit Vector**: **X Component**, **Y Component**, and **Z Component** appear in which you can enter values or specify functions.

Core Loss Model for an RMxprt Material

Specify the following for core loss type and specify the units:

Name	Value
None	No core loss is to be calculated for this material.
Electrical Steel	The following parameters appear: • **Kh** • **Kc**

	• **Ke** **Note** Selecting **Electrical Steel** also enables the Calculate Properties for Core Loss Coefficient pull-down list item at the bottom of the dialog box.
Power Ferrite	The following parameters appear: • **Cm** • **X** • **Y** **Note** Selecting **Power Ferrite** also enables the Calculate Properties for Core Loss versus Frequency pull-down list item at the bottom of the dialog box.

Calculating Properties for Core Loss in RMxprt (BP Curve)

To be able to extract parameters from the loss characteristics (B-P Curve), you first set the **Core Loss Model** of the material to Electrical Steel as a material property in the **View / EditMaterial** window.

To calculate core loss properties for an electrical steel material:

1. Click **Tools>Edit Libraries>Materials**.
   - Or in the project tree, you can also right-click **Materials**, and select **Edit Library**.

   The **Edit Libraries** window appears.
1. Click **Add Material**.

   The **View/Edit Material** window appears.
2. In the **Core Loss Type** row, select **Electrical Steel** from the **Value** pull-down list.

   This enables the **Calculate Properties for** pull-down menu at the bottom of the dialog box with the following two choices:
   - **Calculate Properties for: Core Loss at One Frequency**
   - **Calculate Properties for: Core Loss versus Frequency**

   It also displays parameters associated with Electrical Steel materials.

Electrical Steel Core Loss from a Single-Frequency Loss Curve

With the **Core Loss Type** set to Electrical Steel:

1. Select **Core Loss at One Frequency** from the **Calculate Properties for:** pull-down menu.

   The **BP Curve** window appears.
2. Do one of the following to specify a BP curve:

- Import the curve from a saved file.
- Enter the coordinates manually.

Note	The accuracy in inputting the data for B-P Curve for the electrical steel material has significant effect on the correctness of the analyses to the electromagnetic characteristics of the electric machine. You should input the data for B-P Curve according to the accurate data provided by the manufacturers of materials.

3. Select the units in which the B-P curve is measured from the **Core Loss Unit** pull-down list.
4. Type values and select units for the following:
   - Mass Density
   - Frequency
   - Thickness
   - Conductivity

   The following parameters are dynamically updated with both the specified unit and the standard unit (w/m^3) as the input data changes.
   - **Kh** -Hysteresis
   - **Kc** - Classical Eddy
   - **Ke** - Excess
5. Click **OK**.

   The **View/Edit Material** dialog box reappears. The property values fro **Kh**, **Kc**, and **Ke** are updated as new default values.

Computation of Core Loss from a Single-Frequency Loss Curve

The principles of the computation algorithm are summarized as follows.

The iron-core loss is expressed as:

$$p_v = p_h + p_c + p_e$$

$$= K_1 B_m^2 + K_2 B_m^{1.5}$$

where the eddy-current loss is

$$p_c = k_c (f B_m)^2$$

the hysteresis loss is

$$p_h = k_n f B_m^2$$

and the excessive loss is

$$p_e = k_e (f B_m)^{1.5}$$

Therefore

$$K_1 = k_h f + K_c f^2$$
$$K_2 = k_e f^{1.5}$$

The classical eddy-current loss coefficient is calculated directly as

$$k_c = \pi^2 \sigma \frac{d^2}{\delta}$$

where σ is the conductivity and *d* is the thickness of one lamination sheets.
Minimize the quadratic form to obtain K_1 and K_2.

$$err(K_1, K_2) = \sum_i \left[p_{vi} - \left(K_1 B_{mi}^2 + K_2 B_{mi}^{1.5} \right) \right]^2 = min$$

where P_{vi}, B_{mi} – the i-th point of the data on the loss characteristics curve.

The other two loss coefficients are obtained as

$$k_h = \frac{K_1 - k_c f_0^2}{f_0}$$

$$k_e = \frac{K_2}{f_0^{1.5}}$$

where f_0 is the testing frequency for **B-H Curve**.

Related Topics

Calculating Properties for Core Loss (BP Curve) for Maxwell

Core Loss Coefficients for Electrical Steel

Core Loss Coefficient Extraction

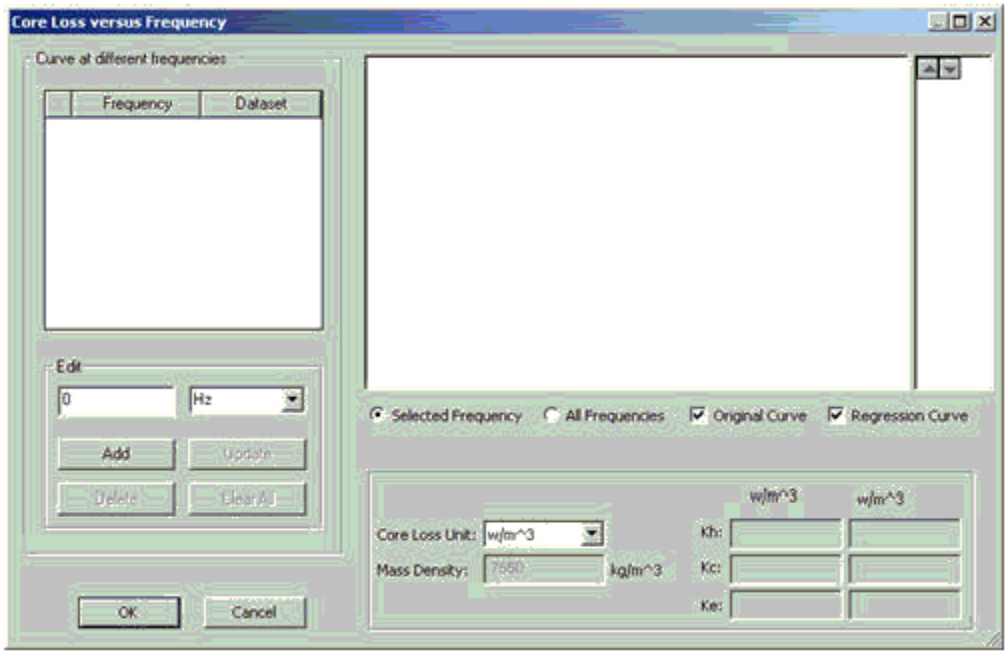
Core Loss Model for a Maxwell Material

Electrical Steel Core Loss from Multi-Frequency Loss Curves

With the **Core Loss Type** set to Electrical Steel:

1. Select **Core Loss versus Frequency** from the **Calculate Properties for:** pull-down menu.

   The **Core Loss versus Frequency** window appears as shown:



2. Using the Edit area, **Add** frequency points at which a dataset is available for the Core Loss.
3. For each Dataset added, click the **Edit Dataset** button to launch the Edit Dataset dialog.
4. Add Dataset information for the frequency by manually entering the data or importing the data from a table. Click **OK** to accept the dataset and return to the **Core Loss versus Frequency** dialog box. Continue adding dataset information until all frequencies have datasets defined.
5. Click **OK** when all frequencies have valid data to complete the core loss calculation and return to the **View/Edit Material** dialog.

Computation of Core Loss from Multi-Frequency Loss Curves

The principles of the computation algorithm are summarized as follows.

The iron-core loss is expressed as:

$$p_v = p_h + p_c + p_e$$

$$= k_h f B_m^2 + k_c (f B_m)^2 + k_e (f B_m)^{1.5}$$

Minimize the quadratic form to obtain k_h , k_c and k_e directly.

$$err(k_h, k_c, k_e) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} \left[p_{vij} - \left(k_h f_i B_{mij}^2 + k_c f_i^2 B_{mij}^2 + k_e f_i^{1.5} B_{mij}^{1.5} \right) \right]^2 = min$$

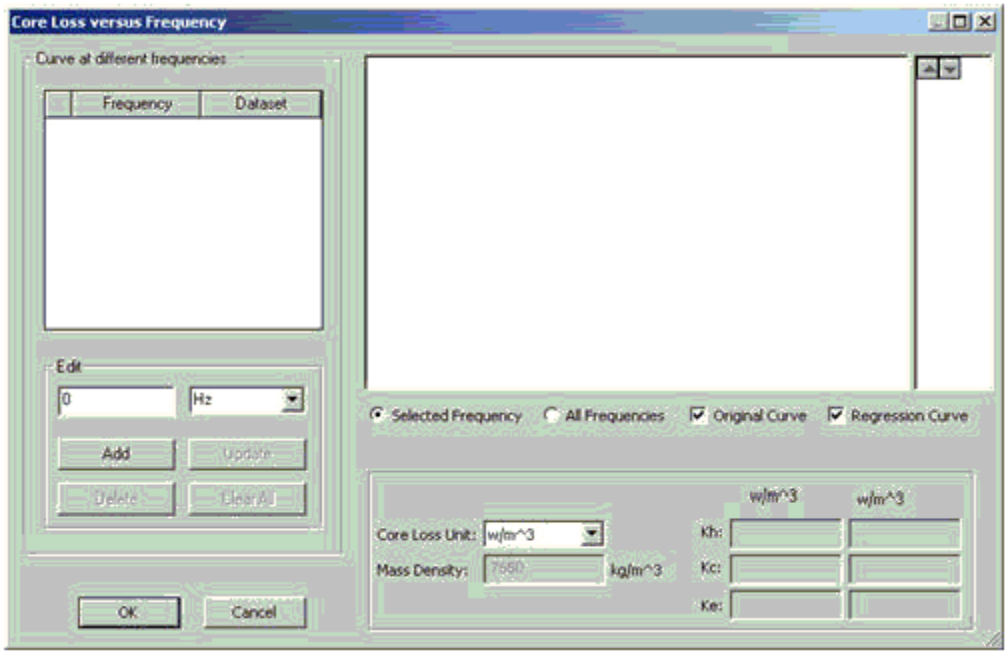
where m - the number of loss curves, n_i - the number of points of the i-th loss curve, and $P_{vij} = f(f_i, B_{mij})$ - two dimensional lookup table for multi-frequency loss curves.

Power Ferrite Core Loss from Multi-Frequency Loss Curves

With the **Core Loss Type** set to Power Ferrite:

1. Select **Core Loss versus Frequency** from the **Calculate Properties for:** pull-down menu.

   The **Core Loss versus Frequency** window appears as shown:



2. Using the Edit area, **Add** frequency points at which a dataset is available for the Core Loss.
3. For each Dataset added, click the **Edit Dataset** button to launch the Edit Dataset dialog.
4. Add Dataset information for the frequency by manually entering the data or importing the data from a table. Click **OK** to accept the dataset and return to the **Core Loss versus Frequency** dialog box. Continue adding dataset information until all frequencies have datasets defined.

5. Click **OK** when all frequencies have valid data to complete the core loss calculation and return to the **View/Edit Material** dialog.

Computation of Power Ferrite Core Loss from Loss Curves

The principles of the computation algorithm are summarized as follows.
The iron-core loss is expressed as:

$$p_v = C_m f^x B_m^y$$

or

$$\log(p_v) = c + x \cdot \log(f) + y \cdot \log(B_m)$$

where

$$c = \log(C_m)$$

Minimize the quadratic form to obtain *c*, *x* and *y*.

$$err(c,x,y) = \sum_{i=1}^{m} \sum_{j=1}^{n_i} [\log(p_{vij}) - (c + x \cdot \log(f_i) + y \cdot \log(B_{mij}))]^2 = min$$

where m - the number of loss curves, n_i - the number of points of the i-th loss curve, and $P_{vij} = f(f_i, B_{mij})$ - two dimensional lookup table for multi-frequency loss curves. Then C_m is calculated from the equation above.

Mass Density for RMxprt Material

Provide a Simple value for Mass density in kg/m^3.

Composition for RMxprt Material

Specify whether the composition is Solid or Lamination.

If Lamination, specify the:

- Stacking Factor - takes a simple value
- Stacking Direction - a drop down menu lets you select V(1), V(2), or V(3).

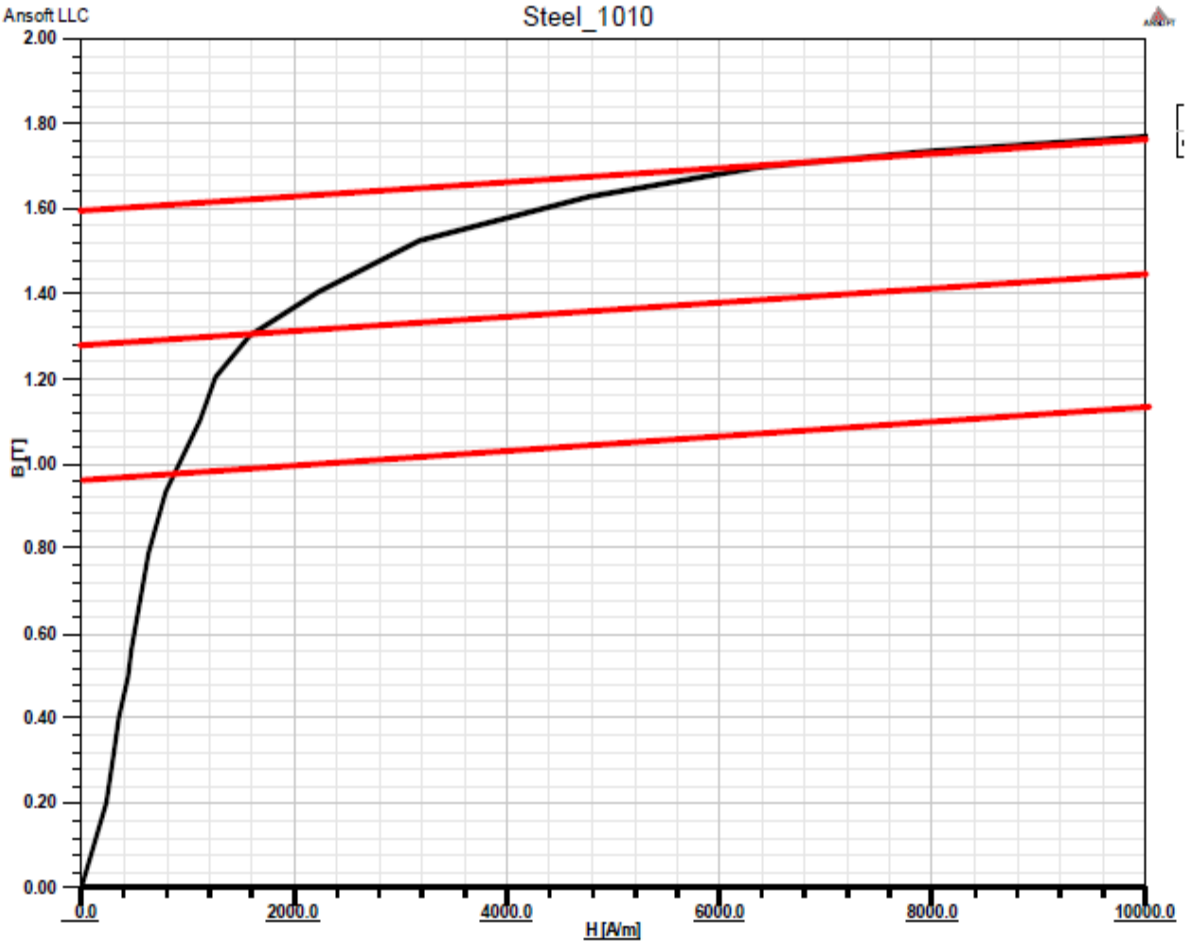
Note	This property is not used in RMxprt design analysis, but it will be transferred to Maxwell 3D Design automatically when the design is created by RMxprt.

Permanent Magnet Materials in RMxprt

A permanent magnet is defined as a material that generates a magnetic flux due to permanent magnetic dipoles in that material.

Compute Remanent Br from B-H curve

The value of the remanent Br of the individual element after the magnetization field is computed is determined in such a way: after having located the operating point on the original non-remanent B-H curve, draw a line which is parallel to the original recoil curve with the slope of $\mu_0\mu_r$ and passes the operating point, the intersection of this line with B-axis is the remanent Br as the result of the applied magnetization field.



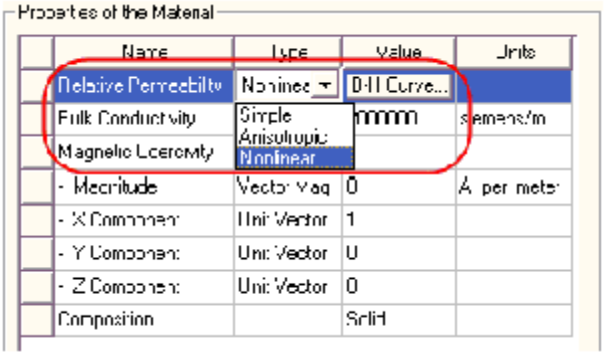
Calculating the Properties for a Non-Linear Permanent Magnet in RMxprt

Non-linear permanent magnet properties may be specified in one of two ways.

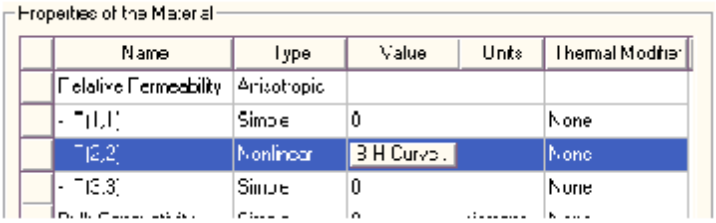
First, a BH curve may be input directly as follows:

1. Click the **View/Edit Materials…** button in the **Edit Libraries** dialog box.

   The **View/Edit Material** dialog box appears.
2. The nonlinear BH curve is defined by setting the **Relative Permeability** **Type** either to **Nonlinear** or **Anisotropic.**



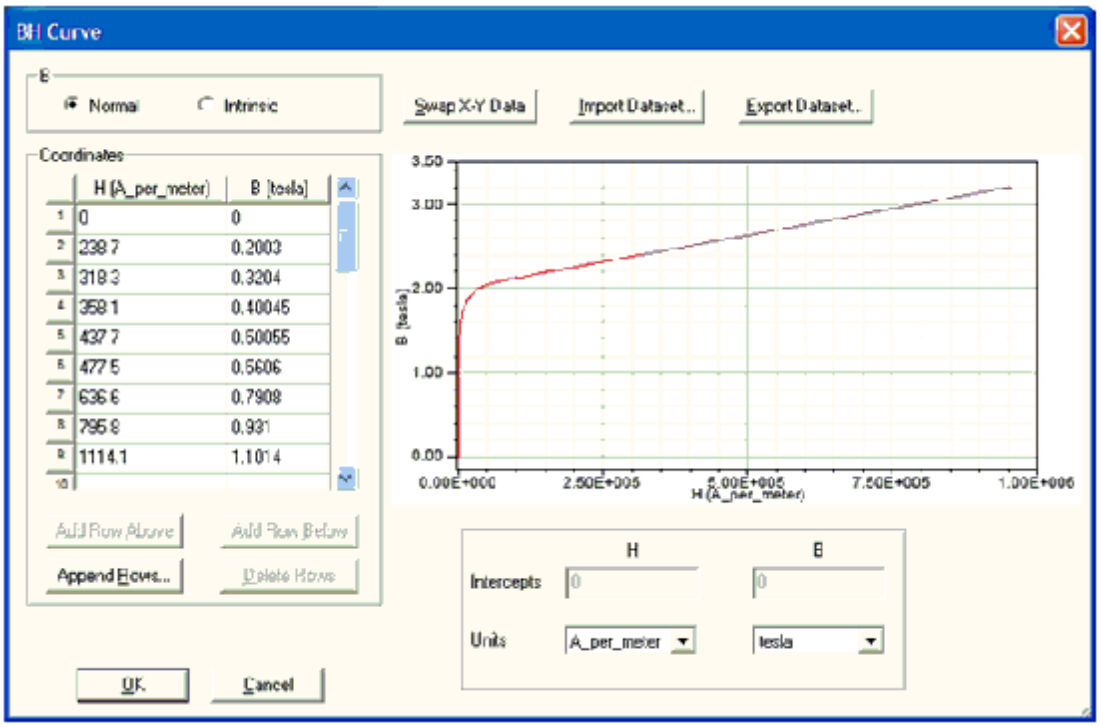
If you select Anisotropic, each of its components can be selected **Nonlinear** and can be specified by a **BH Curve**.



A **B-H Curve** button appears in the nonlinear property's **Value** column

3. Input the BH curve by clicking the **B-H Curve** button in the property **Value** column.

   This opens the **BH Curve** dialog box in which you can input (or modify) curve data. (Refer to *Adding Datasets* for general information on working with datasets.



- For a **Normal** BH curve, the slope of the curve can not be less than that of free space anywhere along the curve.
- For an **Intrinsic** BH curve, the slope of the curve can not be less than 0.

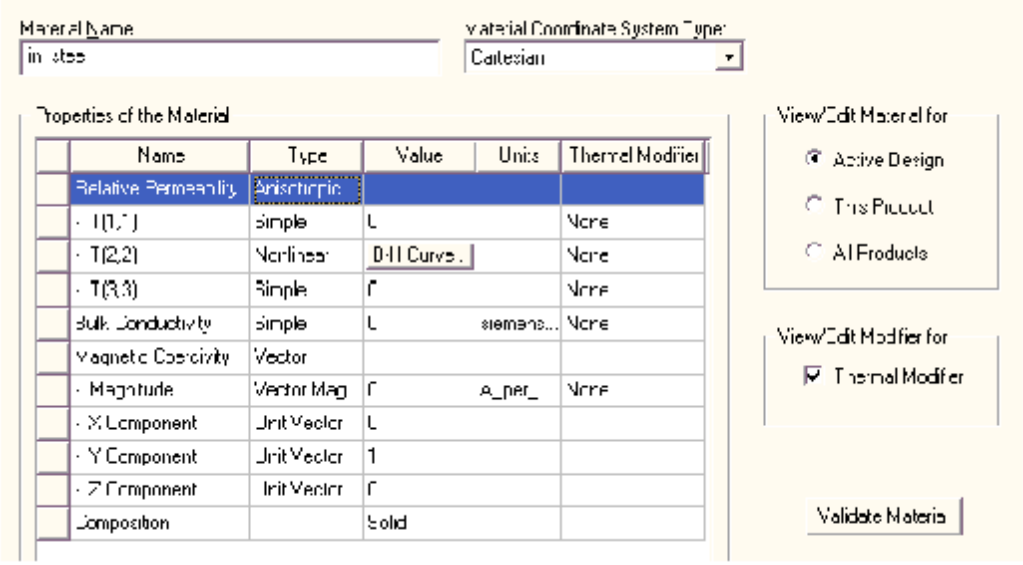
Note	• The **Intrinsic** BH curve is supported only in Maxwell 2D/3D magnetostatic and transient design types. A material property defined using an Intrinsic BH curve will fail validation check in all the other product/design types. • When an **Intrinsic** BH curve is added, the **Relative Permeability Value** button label changes to **Bi-H Curve** as visual indication of the type of curve currently defined for the material.

When you **OK** the dialog box, an error message displays if a slope is out of tolerance, identifying the data points between which the slope is less than that of free space.

The operations to input a nonlinear demagnetization curve are the same as entering a BH curve for Steel material. When a BH curve goes through the second quadrant or third quadrant, the curve is treated as a demagnetization curve.

4. To model *temperature dependency* for a nonlinear permanent magnet you must:
   - Use an **Intrinsic** BH curve to model the **Relative Permeability**
   - Specify a Thermal Modifier for both **Relative Permeability** and the **Magnitude** of **Magnetic Coercivity**. Apply a thermal Modifier by selecting the **Thermal Modifier** check box. Checking this box causes the **Thermal Modifier** column to display at the right side of the **Properties of the Material** table. Selecting **Edit** rather than **None** causes display

of the **Edit Thermal Modifier** dialog.



Alternatively, a non-linear BH curve can be modeled by the following four parameters:

- residual flux density B_r
- coercive field force H_c
- maximum energy product $(BH)_{max}$
- relative recoil permeability μ_r

From the V**iew/Edit Materials** window:

1. Set the **Relative Permeability** to **Nonlinear**.

   This enables the **Calculate Properties for...** drop down menu at the bottom of the window.
2. Click **Non-Linear Permanent Magnet** from the drop down menu.

   This displays the **Properties for Non-Linear Permanent Magnet** dialog box, which contains the following fields into which you enter the appropriate values.

Mur	Provide a value for relative permeability.
Hc	Coercive field force H_c in the units specified. Provide a value and select units from the drop down menu.
Br	Residual flux density B_r, in Tesla. If enabled, provide a value and select units from the drop down menu.
BH max	Maximum magnetic energy product $(BH)_{max}$ If enabled, provide a value and select units form the drop down menu.

3. Click **OK** to close the dialog and return to the **View/Edit Materials** window.

The values for **Relative Permeability** and **Magnitude** under **Magnetic Coercivity** are updated as new default values. Maxwell will create a lookup table based on the Four-Parameter Curve Fitting algorithm, and update the coordinates of the BH curve automatically as long as the input data of four parameters pass the validation check.

Related Topics

Non-Linear vs. Linear Permanent Magnets
Calculating the Properties for a Linear Permanent Magnet
Temperature Dependent Nonlinear Permanent Magnets
Irreversible Demagnetization Due to Temperature Change

Calculating the Properties for a Linear Permanent Magnet

Edit a linear demagnetization curve is simple. From the V**iew/Edit Materials** window:

1. Set the **Relative Permeability** to **Simple**.
   This enables the **Calculate Properties for Permanent Magnet** drop down menu at the bottom of the window.
2. Click **Permanent Magnet** from the drop down menu.
   This displays the **Properties for Permanent Magnet** window. This contains the following fields.

Mu (enabled by default)	Provide a value.
Hc (enabled by default)	Coercive field force H_c in the units specified. Provide a value and select units from the drop down menu.
Br/Mp (disabled by default)	Checking this enables the radio buttons that let you specify either Br or Mp.
Br	Residual flux density B_r, in Tesla. If enabled, provide a value and select units from the drop down menu.
Mp	If enabled, provide a value and select units form the drop down menu.

3. Click **OK** to close the dialog and return to the **View/Edit Materials** window.

The values for **Relative Permeability** and **Magnitude** under **Magnetic Coercivity** are updated as new default values.

Note	The accuracy in inputting the characteristic parameters for the permanent-magnetic material has significant effect on the correctness of the analyses to the electromagnetic characteristics of the electric machine. It is suggested that users should input the characteristic parameters of the permanent-magnetic material according to the accurate data provided by the manufacturers of materials. **RMxprt** provides a few

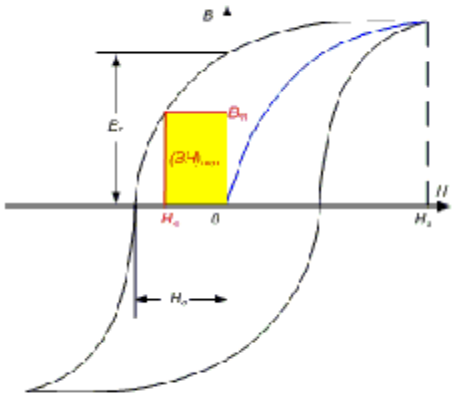
	characteristic parameters of permanent-magnetic materials for reference.

Using Demagnetization Curves

Many permanent magnet manufactures directly provide demagnetization curves for their products, but in most cases, manufactures provide some main parameters, such as residual flux density *Br*, coercive field force *Hc* and maximum magnetic energy product $(BH)_{max}$, and relative recoil magnetic permeability μ_r. This section and the next section describe the basic parameters for the demagnetization curve of permanent magnets and the curve fitting based on these parameters.

Hysteresis Loop

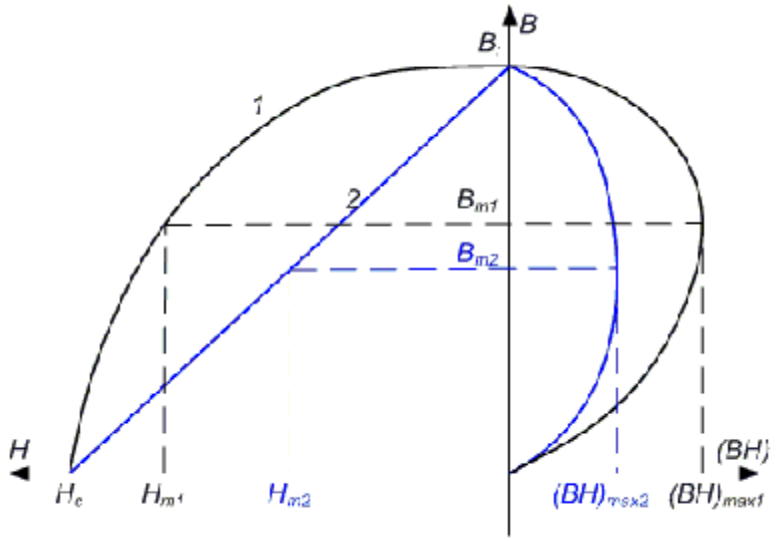
The permanent-magnetic material belongs to hard-magnetic material. It is characterized with "fat" hysteresis loop, which encloses large area as shown in the figure. When magnetized, it keeps high magnetic property with the external magnetic field removed, therefore is used in the permanent-magnet electric machine to produce magnetic field. The characteristics of the permanent-magnetic material are represented with its main parameters: residual flux density B_r, coercive field force H_c and maximum magnetic energy product $(BH)_{max}$.



Demagnetization Curve

The part of the maximum hysteresis loop of the permanent-magnetic material in the second quadrant is called the demagnetization curve as shown in the next figure. It is the basic characteristics curve of the permanent-magnetic material. On the demagnetization curve, the magnetic flux density has positive values, but the magnetic field intensity has negative values. It means that the permanent-magnetic material is applied with demagnetization magnetic field

intensity. Since H_m and B_m are in opposite directions, when the magnetic flux goes through the permanent-magnetic material, the magnetic potential difference along the direction of the magnetic flux does not drop, but rises. Therefore, the permanent-magnetic material is a magnetic source, similar to the electric source in the electric circuit.

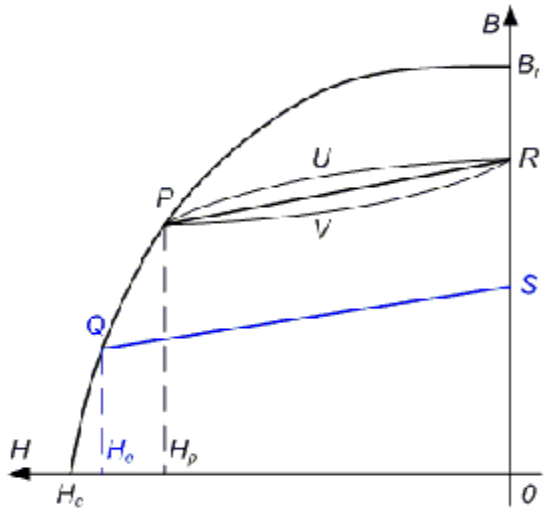


The two extreme positions on the demagnetization curve are the two significant parameters to represent the magnetic characteristics of the permanent-magnetic material. On the demagnetization curve, the value of the magnetic flux density corresponding to zero magnetic field intensity H is termed residual flux density B_r, on the other hand, the value of the magnetic field intensity corresponding to zero magnetic flux density B is termed coercive field force H_c. The produce of the magnetic flux density and the magnetic field intensity at any point on the demagnetization curve is termed magnetic energy product (BH), which is proportional to the magnetic energy density possessed by the permanent magnet at the given operating situation. At the two extreme positions ($B = B_r$, $H = 0$) and ($B = 0$, $H = H_c$), the magnetic energy product is equal to zero. Somewhere at an intermediate position, the magnetic energy reaches its maximum value and is termed maximum magnetic energy product $(BH)_{max}$, which is another significant parameter to represent the magnetic characteristics of the permanent-magnetic material. To some permanent-magnetic materials with linear demagnetization curve, it is obvious that at ($B = B_r / 2$, $H = H_c / 2$), the magnetic energy product reaches its maximum value, i.e. $(BH)_{max} = B_r H_c / 4$.

Recoil Lines

The relationship between the magnetic flux density and the magnetic field intensity represented by the demagnetization curve only exists when the magnetic field intensity varies in the same direction. In fact, when the permanent magnet electric machine is working, the demagnetization field intensity varies repeatedly in both directions. When demagnetization field is applied to the

magnetized permanent magnet, the magnetic flux density decreases along the curly segment B_rP on the demagnetization curve as shown in the figure



If the external demagnetization field intensity H_p is removed when the magnetic flux density reaches the point *P*, the magnetic flux density will increase not along the original demagnetization curve, but along another curve *PVR*. If the external demagnetization field intensity is reapplied, the magnetic flux density will decrease along the new curve *RUP*. By repeatedly applying the demagnetization field intensity, a localized loop is formed and is termed local hysteresis loop. On the local hysteresis loop, the rising segment and the dropping segment are quite close to each other, therefore can be approximated by the straight line *PR*, which is termed recoil line with *P* as the starting point. If demagnetization field with intensity H_q not exceeding the original value H_p is applied thereafter, the magnetic flux density will vary reversibly along the recoil line *PR*. If $H_q > H_p$, the magnetic flux density drops to a new starting point *Q*. It will vary along the new recoil line *QS*, but not the previous one *PR*. This sort of irreversible variation in magnetic flux density causes instability in the characteristics of electric machines and complicates the design computation of permanent magnet electric machines, therefore should be avoided as possible.

Recoil Magnetic Permeability

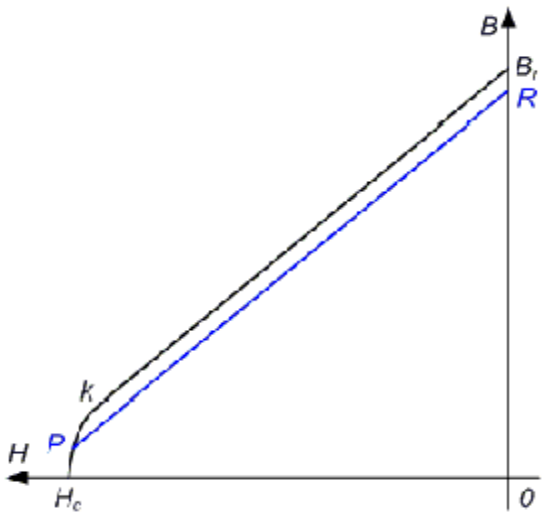
The ratio of the average slope of the recoil line to the magnetic permeability in vacuum μ_0 ($\mu_0 = 4 \times 10^{-7}$ H/m) is termed relative recoil magnetic permeability or recoil magnetic permeability for short, μ_r:

$$\mu_r = \frac{1}{\mu_0}\left|\frac{\Delta B}{\Delta H}\right|$$

If the demagnetization curve is curly, the value of μ_r depends on the location of the starting point and is a variable, but typically varies within a small range. Therefore, it is approximated as a constant and is taken as the slop of the tangent to the point (B_r, 0) on the demagnetization curve. In other words, the recoil lines at different starting points are approximated as a family of parallel lines, which are all parallel to the tangent to the point (B_r, 0) on the demagnetization curve.

Inflection Point

Some permanent-magnetic materials, such as some ferrite permanent-magnetic materials, show straight upper segment on the demagnetization curve. When the demagnetization field intensity drops to a given value, the demagnetization curve turns to decrease rapidly. The turning point is termed inflection point. If the demagnetization field intensity does not exceed the inflection point *k*, the recoil line coincides with the straight segment on the demagnetization curve. If the demagnetization field intensity exceeds the inflection point *k*, the new recoil line *RP* no longer coincides with the straight segment on the demagnetization curve.



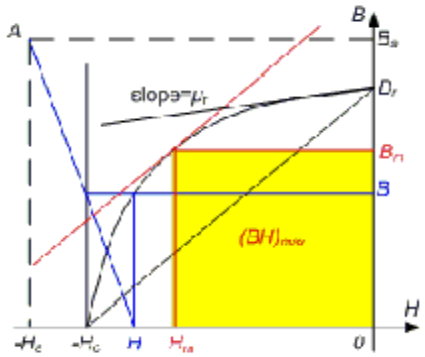
Some permanent-magnetic materials, such as most of the rare-earth permanent-magnetic materials, show straight demagnetization curve in the whole range. In those cases, the recoil line coincides with the demagnetization curve. This makes the magnetic property keep stable while the permanent-magnet electric machine is working. This is the best ideal demagnetization curve for electric machine applications.

Curve Fitting of Demagnetization Curves

RMxprt fits the demagnetization curve according to the given characteristic parameters B_r, H_c, $(BH)_{max}$ and μ_r.

Three Parameter Curve Fitting

Given the three characteristic parameters *Br*, *Hc* and $(BH)_{max}$, the principles of the three-parameter curve fitting algorithm are summarized as follows.Using the following figure as a reference:



$$H_a = \frac{H_c}{a}$$

and

$$B_a = \frac{B_r}{a}$$

where a<1.

Any magnetic flux density *B* in the interval

$$0 \le B \le B_r$$

corresponds to the magnetic field intensity *H*:

$$H = -H_c + \frac{H_a - H_c}{B_a - B} B = -H_c + \frac{H_c(1-a)}{B_r - aB} B = -H_c \frac{B_r - B}{B_r - aB}$$

The tangent at any point is given by:

$$\frac{dB}{dH} = \left(\frac{dH}{dB}\right)^{-1} = \frac{\left(1 - a\frac{B}{B_r}\right)^2}{1-a} \frac{B_r}{H_c}$$

The magnetic flux density B_m and the magnetic field intensity H_m corresponding to the maximum magnetic energy product satisfy the following relationship:

$$\left.\frac{dB}{dH}\right|_{B = B_m} = \frac{B_r}{H_c}$$

Solving yields:

$$B_m = \frac{B_r}{1 + \sqrt{1-a}} \qquad and \qquad H_m = -\frac{H_c}{1 + \sqrt{1-a}}$$

Let the magnetic energy product at the point equal to $(BH)_{max}$ be:

$$\left|B_m H_m\right| = \frac{B_r H_c}{(1 + \sqrt{1-a})^2} = (BH)_{max}$$

Solving yields:

$$a = 2\sqrt{\left|\frac{B_r H_c}{(BH)_{max}}\right|} - \left|\frac{B_r H_c}{(BH)_{max}}\right|$$

The relative recoil magnetic permeability μ_r is calculated as:

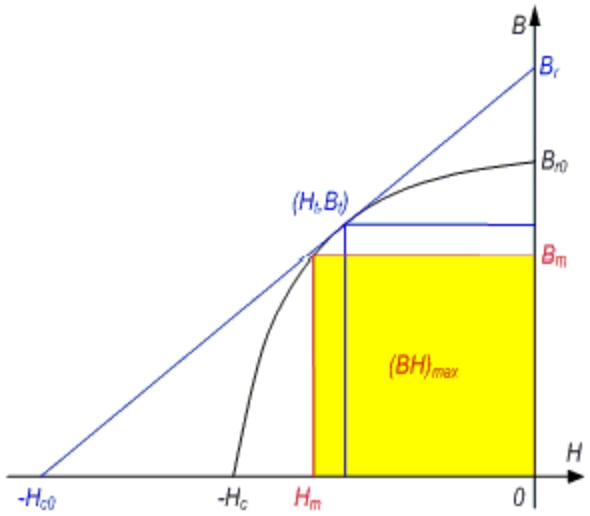
$$\mu_r = \frac{1}{\mu_0}\frac{dB}{dH}\bigg|_{B=B_r} = (1-a)\frac{B_r}{\mu_0 H_c}$$

Four Parameter Curve Fitting

The three-parameter curve fitting technique fits the demagnetization curve well. For the nonlinear permanent-magnetic material, the real operating point lies often not on the demagnetization curve, but on the recoil line. The relative recoil magnetic permeability calculated with the three-parameter curve fitting technique will cause deviation, therefore RMxprt employs a more accurate fitting technique: four-parameter curve fitting technique, as introduced below.

Given the four characteristic parameters B_r, H_c, $(BH)_{max}$ and μ_r, the principles of the four-parameter curve fitting algorithm are summarized as follows:

1. Draw a line through the point (0, B_r) with the slope equal to $-\mu_r\mu_o$ as shown in the Figure. The segment of this line in the second quadrant is termed the ideal recoil line.



2. Find the virtual magnetic flux density B_{r0}.
3. With B_{r0}, H_c, and $(BH)_{max}$, draw the demagnetization curve with the three-parameter curve fitting technique. The curve should touch the ideal recoil line at the tangent point (H_t, B_t).
4. Any magnetic flux density B in the interval

$$0 \le B \le B_r$$

corresponds to the magnetic field intensity H:

$$H = \begin{cases} -H_c \dfrac{B_{r0} - B}{B_{r0} - a_o B} & B \le B_t \\ H_t + \dfrac{B - B_t}{\mu_r \mu_o} & B \ge B_t \end{cases}$$

The virtual magnetic flux density B_{r0} is found by iteration:

1. Start from the initial guess for the lower and the upper bounds for the virtual magnetic flux density B_{r0}:

$$B_0 = max\left(\mu_r \mu_o H_c, \frac{(BH)_{max}}{H_c}\right)$$

$$B_1 = B_r$$

2. Let:

$$B_{r0} = \frac{B_0 + B_1}{2}$$

3. With B_{r0}, H_c, and $(BH)_{max}$, draw the demagnetization curve with the three-parameter curve fitting technique.

$$a_0 = 2\sqrt{\left|\frac{B_{r0}H_c}{(BH)_{max}}\right|} - \left|\frac{B_{r0}H_c}{(BH)_{max}}\right|$$

4. The curve should touch a line parallel to the ideal recoil line at the tangent point (H_t, B_t).

$$B_t = \begin{cases} B_{r0} & a_0 = 0 \\ \dfrac{B_{r0} - \sqrt{\mu_r \mu_0 H_c (1 - a_0) B_{r0}}}{a_0} & a_0 > 0 \end{cases}$$

and

$$H_t = -H_c \frac{B_{r0} - B_t}{B_{r0} - a_0 B_t}$$

1. For any magnetic flux density B in the interval

$$0 \le B \le B_r$$

the corresponding magnetic field intensity H will be calculated by:

$$H = \begin{cases} -H_c \dfrac{B_{r0} - B}{B_{r0} - a_o B} & B \le B_t \\ H_t + \dfrac{B - B_t}{\mu_r \mu_o} & B \ge B_t \end{cases}$$

1. Calculate the value of H_r corresponding to B_r using:

$$H_r = H_t + \frac{B_r - B_t}{\mu_r \mu_0}$$

2. If H_r>0, the assumed virtual B_{r0} is too small, the lower bound of the interval needs to be increased, so let B_0=B_{r0}. If, however, H_r<0, the assumed B_{r0} is too big, the upper bound of the interval needs to be decreased, so let B_1=B_{r0}.
3. Repeat steps (2) through (7) until H_r converges to 0 within satisfactory precision.

Conductor Data

When a material is assigned to a machine part, RMxprt checks that the material is appropriate for the machine part based upon whether it is a conductor or other material type. RMxprt distinguishes conductors based on material threshold settings. RMxprt's treatment of conductors may be set by the used by adjusting the **Material Threshold**.

Setting the Material Threshold for RMxprt

1. Click **RMxprt>Design Settings.**

   The **Design Settings** dialog box appears with the **Material Threshold** tab selected.
2. Type a value in the **Conductivity Threshold** text box (Default=100,000).
3. Type a value in the **Permeability** text box (Default=100).

Note	RMxprt will treat materials with conductivity greater than 100,000 as conductors, and materials with Permeability greater than 100 as steels.

4. If you want these values to be the default, change the values by clicking the **Tools>Options>RMxprt Options** menu and setting the material thresholds in the **RMxprt Options** dialog.
5. Click **OK**.

Editing Conductivity Properties in RMxprt

1. Click **Tools>Edit Libraries>Materials** to open the **Edit Libraries** dialog box.
2. Select the material in the list whose conductivity properties you wish to edit and click the **View/Edit Materials** button.
3. In the **View/Edit Materials** dialog, **Bulk Conductivity** has two property types in the **Type** pull-down list.

- **Simple**: For this type, you must enter a simple value for the property value.
- **Anisotropic**: For this type, you must specify material properties for three principal directions:

a. **T(1,1)**

b. **T(2,2)**

c. **T(3,3)**

Note	The Anisotropic type is not used in RMxprt design analysis, but it will be transferred to Maxwell 3D Design automatically when the design is created by RMxprt.

4. **Mass Density** is a Simple parameter.

Specifying RMxprt Solution Settings

Specify how RMxprt computes a solution by adding a *solution setup* to the design. You can define more than one solution setup per design.

To add a solution setup to a design:

1. Select a design in the project tree.
2. Click **RMxprt**>**Analysis Setup>Add Solution Setup**.
   - Alternatively, right-click **Analysis** in the project tree, and then click **Add Solution Setup** on the shortcut menu.

   The **Solution Setup** dialog box appears. It is divided among the following tabs:

General	Includes general solution settings, including rated output power, speed, operating temperature, etc.
<machine type>	Includes settings specific to the selected machine type.
Defaults	Includes settings to save and clear user-defined default values.

3. Click the**General** tab.
4. If available for the machine you are using, select an **Operation Type** from the pull-down list.

   This may be Motor or Generator.

Note	To enable selection of **Generator** for Adjust-Speed Synchronous Machines, the machine **Control Type** must be set to **AC** in its Properties window.

   When you make the selection, this makes a difference in the **Load Type** available.
5. Select a **Load Type** from the pull-down list. For **Motor** operation, the options are:
   - **Const Speed**
   - **Const Power**
   - **Const Torque**
   - **Linear Torque**
   - **Fan Load**

   If the model has an **Operation Type**, and you select **Generator**, the **Load Type** options are:
   - **Infinite Bus**
   - **Independent Generato**r
6. Type the **Rated Output Power**, and select the units.
7. Type the **Rated Voltage**, and select the units.
8. Type the **Rated Speed**, and select the units.
9. Type the **Operating Temperature**, and select the units.
10. Click the **<machine type>** tab (if any for this machine).

The options vary depending on the machine. For example, the 3 Phase Induction Motor includes options for:

- Frequency and Units
- Winding Connection (Wye or Delta).

The 3 Phase Synchronous Machine includes options for:

- Rated Power Factor
- Winding Connection (Wye or Delta)
- Exciter Efficiency
- Input Exciting current and units

The Brushless DC Motor does not use the <**machine type**> tab.

11. Specify the desired settings, based on the machine type you have selected.
12. Click **OK**.

Note	To edit a setup after it has been created, right-click the specific setup (for example, **Setup1**), under **Analysis** in the project tree, and then click **Properties** on the shortcut menu.

Generating a Custom Design Sheet for RMxprt

RMxprt allows users to import all the data items of **Design Output** into a **Microsoft Excel** worksheet in order for users to design **Design Sheet** of their own styles according to their own requirements using **Microsoft Excel**.

Before you can specify a design sheet, you first need to customize a template in Microsoft Excel and set the export options.

To set the export options:

1. Click **RMxprt>Design Settings**.
   The **Design Settings** dialog box appears. Select the **Export Options** tab.
2. In the **Design Sheet** section, specify an **Excel Template** by clicking the **...** button, selecting the template you want to use, and clicking **Save**.
3. Click **OK**.

To create a design sheet based on a previously-customized template after the design has been analyzed:

1. Click **RMxprt>Analysis Setup>Export**.
   The **Export** dialog box appears.
2. Select **Customized Design Sheet** from the **Type** pull-down menu.
3. Choose the **Solution Setup** from the drop-down menu.
4. In the **Path** field, enter the location to export the files to, or use the ellipsis (...) button to browse to the desired location.
5. Click **OK**.

RMxprt will connect to **Microsoft Excel** according to the set path and automatically import the relevant data from **Design Output** into a copy of the customized **Design Sheet Template**.

Specific data not available in **Design Output** is shown as **N/A**, for instance, **RS** in **Slot Sizes** is *N/A* **(mm)** when stop type is 1. The post-processing of data is automatically performed, for instance, **Winding Weight** is *0.947* **(kg)**.

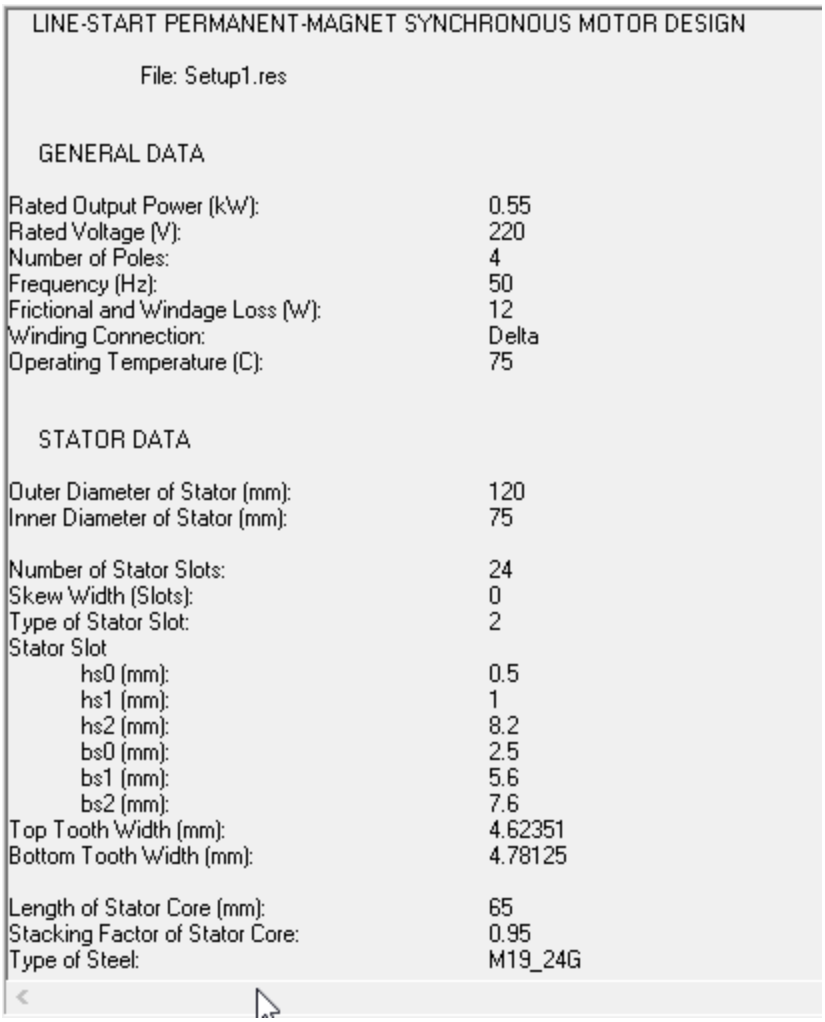
Note	Several **Design Sheet** examples are shipped with RMxprt in the examples subdirectory of your installation.

Related Topics

Exporting a Maxwell or Simplorer Model

Key Words in Output Data for RMxprt

In **Design Output** of **RMxprt**, the literal expressions for various parameters are termed key words of output data. For example, the **Design Output** for Line-start Permanent-magnet Synchronous Motor (*lssm*) example project sm-1.aedt is shown below.



LINE-START PERMANENT-MAGNET SYNCHRONOUS MOTOR DESIGN

File: Setup1.res

GENERAL DATA

Rated Output Power (kW):	0.55
Rated Voltage (V):	220
Number of Poles:	4
Frequency (Hz):	50
Frictional and Windage Loss (W):	12
Winding Connection:	Delta
Operating Temperature (C):	75

STATOR DATA

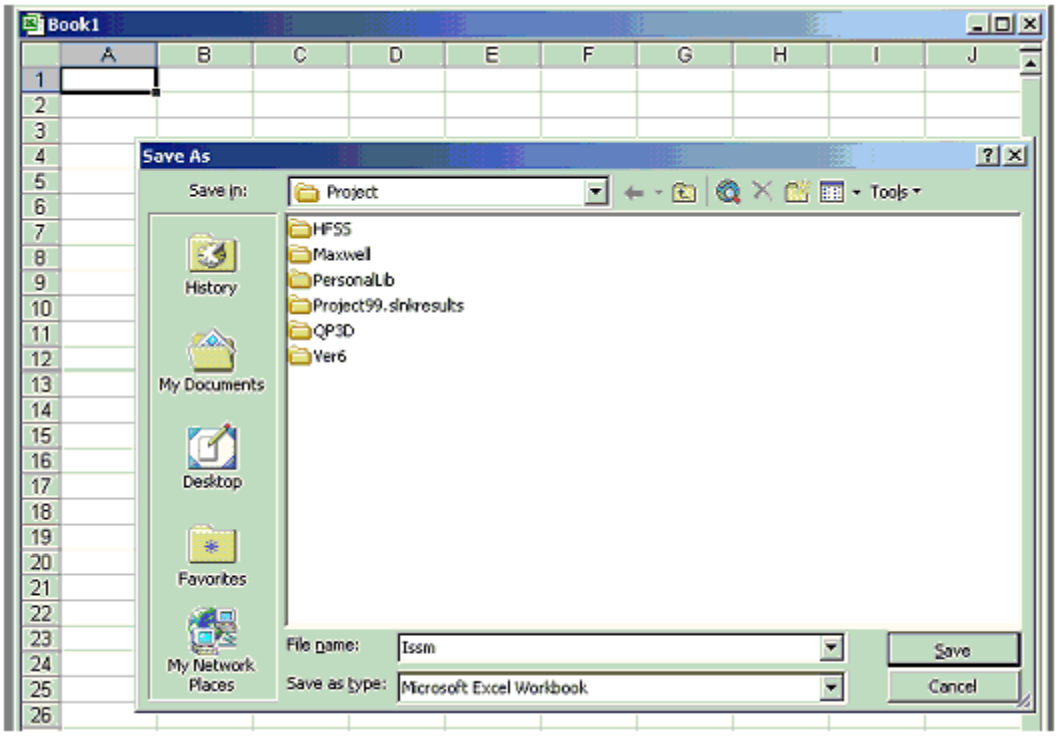
Outer Diameter of Stator (mm):	120
Inner Diameter of Stator (mm):	75
Number of Stator Slots:	24
Skew Width (Slots):	0
Type of Stator Slot:	2
Stator Slot	
hs0 (mm):	0.5
hs1 (mm):	1
hs2 (mm):	8.2
bs0 (mm):	2.5
bs1 (mm):	5.6
bs2 (mm):	7.6
Top Tooth Width (mm):	4.62351
Bottom Tooth Width (mm):	4.78125
Length of Stator Core (mm):	65
Stacking Factor of Stator Core:	0.95
Type of Steel:	M19_24G

In the **Design Sheet**, "Rated Output Power", "Rated Voltage", "Number of Poles", "Frequency", "Frictional and Windage Loss", etc. are all key words of output data.

Creating RMxprt Customized Design Sheet Template

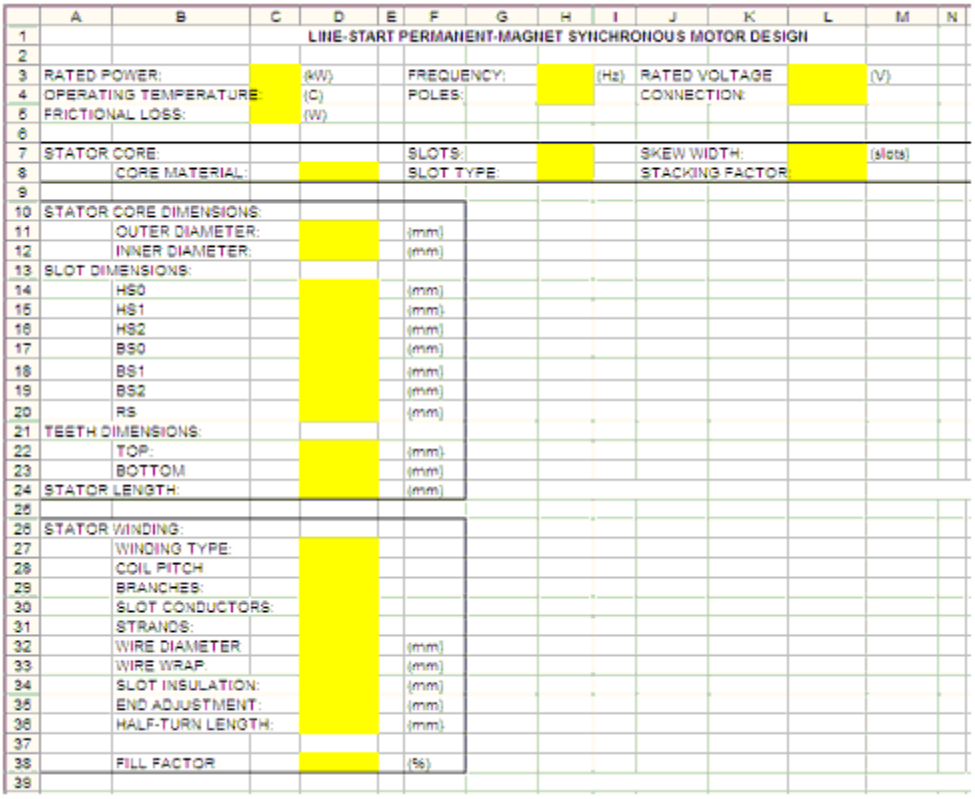
As a sample example, the Line-start Permanent-magnet Synchronous Motor (*lssm*) is used to demonstrate the process for creating a **CustomizedDesign Sheet template**.

Start **Microsoft Excel**, rename the blank worksheet file as "*lssm*" and save it under the path *ansoft\rmxprt5* as shown below.



Design Template of Microsoft Excel Worksheet in Preferred Styles

According to special requirements and preferences, select relevant parameters, arrange the page formats of worksheet. Taking as example a part of the parameters of the **Design Sheet** of the Line-start Permanent-magnet Synchronous Motor (*lssm*), the designed worksheet template is shown below.

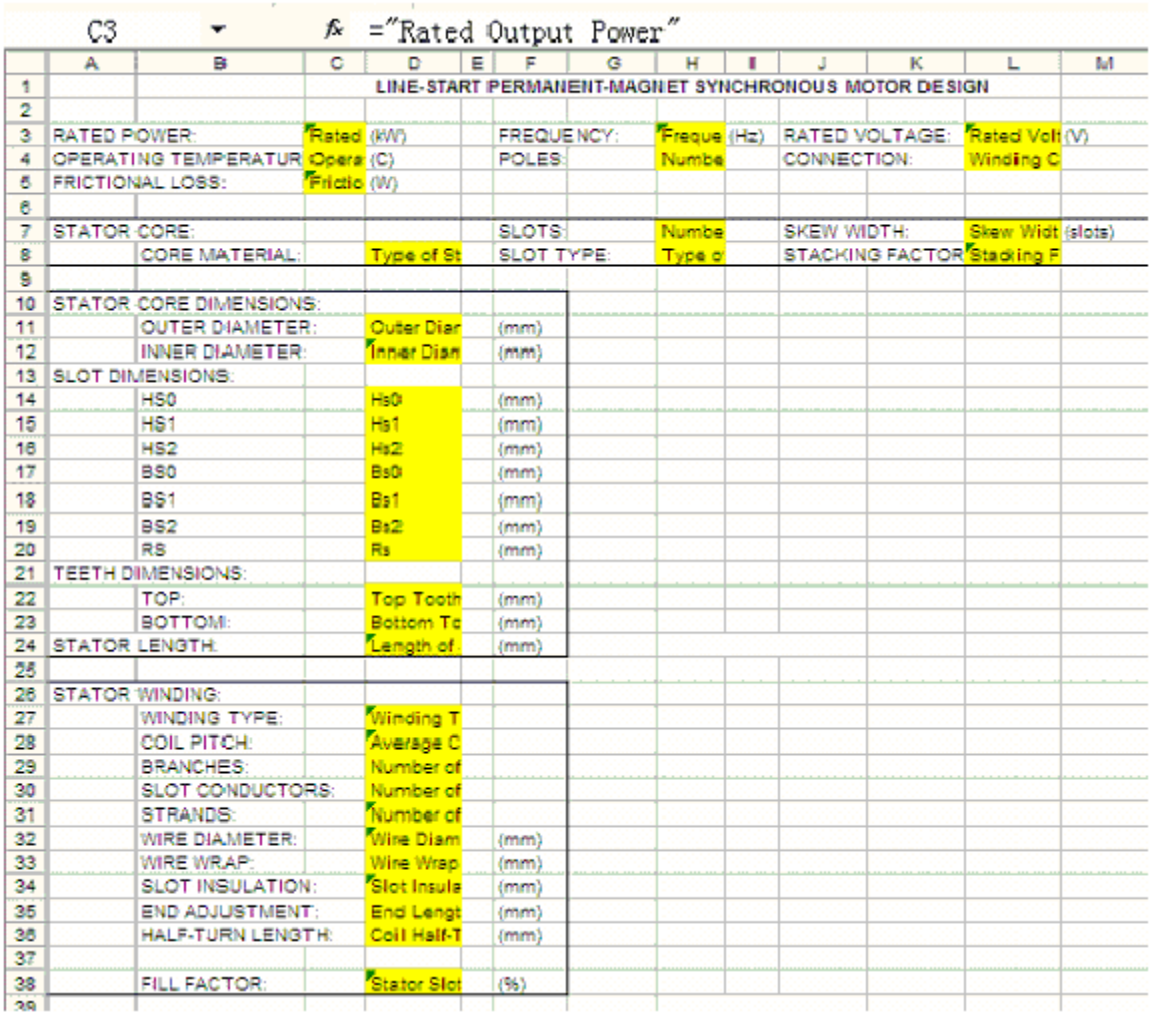


	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1				LINE-START PERMANENT-MAGNET SYNCHRONOUS MOTOR DESIGN										
2														
3	RATED POWER:			(kW)		FREQUENCY:			(Hz)	RATED VOLTAGE			(V)	
4	OPERATING TEMPERATURE:			(C)		POLES:				CONNECTION:				
5	FRICTIONAL LOSS:			(W)										
6														
7	STATOR CORE:					SLOTS:				SKEW WIDTH:			(slots)	
8		CORE MATERIAL:				SLOT TYPE:				STACKING FACTOR:				
9														
10	STATOR CORE DIMENSIONS:													
11		OUTER DIAMETER:				(mm)								
12		INNER DIAMETER:				(mm)								
13	SLOT DIMENSIONS:													
14		HS0				(mm)								
15		HS1				(mm)								
16		HS2				(mm)								
17		BS0				(mm)								
18		BS1				(mm)								
19		BS2				(mm)								
20		RS				(mm)								
21	TEETH DIMENSIONS:													
22		TOP:				(mm)								
23		BOTTOM				(mm)								
24	STATOR LENGTH:					(mm)								
25														
26	STATOR WINDING:													
27		WINDING TYPE:												
28		COIL PITCH												
29		BRANCHES:												
30		SLOT CONDUCTORS:												
31		STRANDS:												
32		WIRE DIAMETER				(mm)								
33		WIRE WRAP				(mm)								
34		SLOT INSULATION:				(mm)								
35		END ADJUSTMENT:				(mm)								
36		HALF-TURN LENGTH:				(mm)								
37														
38		FILL FACTOR				(%)								
39														

In the template, the yellow-colored areas are reserved for importing data.

Resort to Key Words in Design Output

In the spaces for importing data in the template (shown in yellow), key in = *"xxxxx"*. Within the double quotation marks, *xxxxx* stands for the relevant key words as shown below.

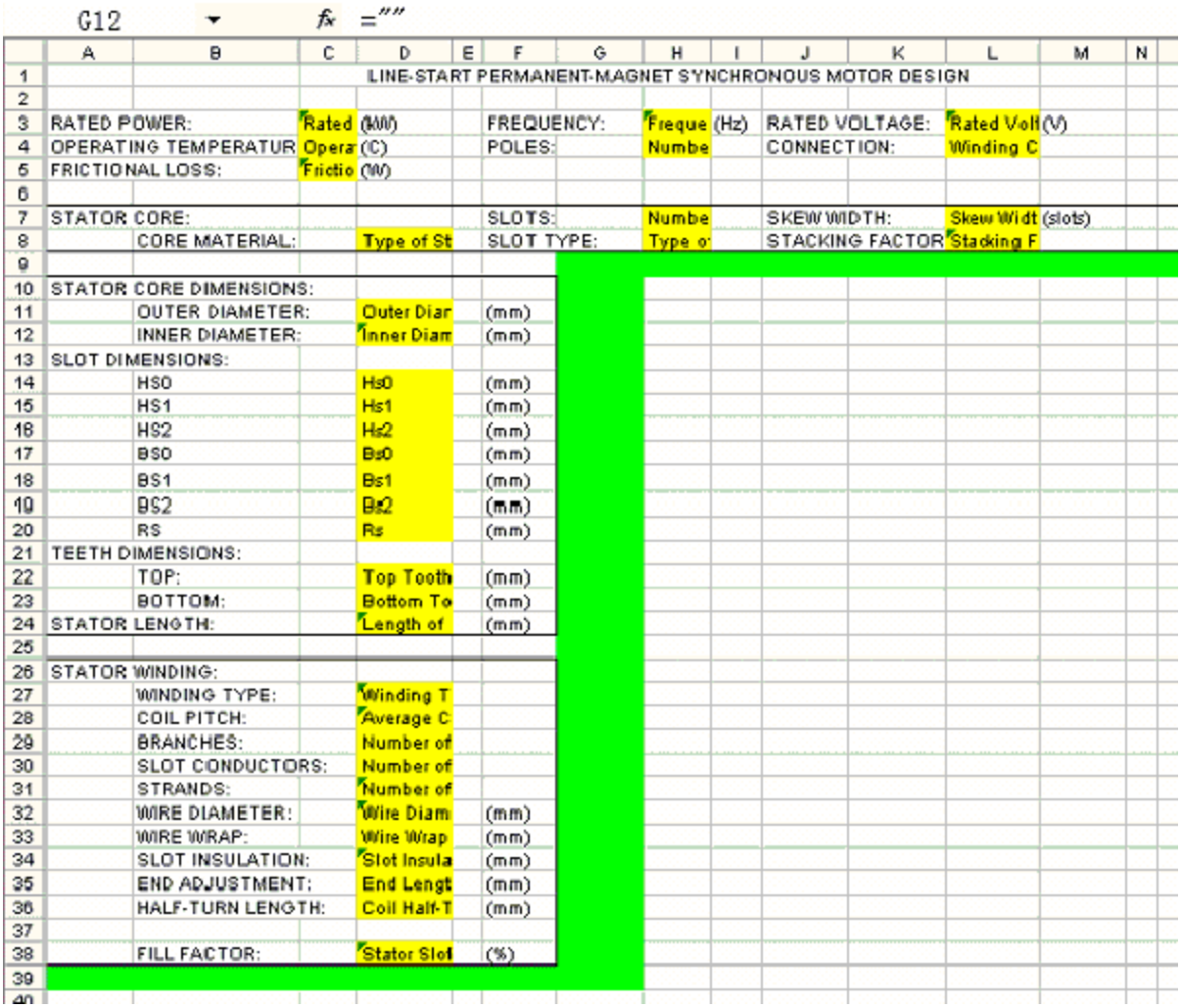


C3 =”Rated Output Power”

	A	B	C	D	E	F	G	H	I	J	K	L	M
1				LINE-START PERMANENT-MAGNET SYNCHRONOUS MOTOR DESIGN									
2													
3	RATED POWER:		Rated	(kW)		FREQUENCY:		Freque	(Hz)	RATED VOLTAGE:		Rated Volt	(V)
4	OPERATING TEMPERATUR		Opera	(C)		POLES:		Numbe		CONNECTION:		Winding C	
5	FRICTIONAL LOSS:		Frictio	(W)									
6													
7	STATOR CORE:					SLOTS:		Numbe		SKEW WIDTH:		Skew Widt	(slots)
8		CORE MATERIAL:		Type of St		SLOT TYPE:		Type o		STACKING FACTOR		Stacking F	
9													
10	STATOR CORE DIMENSIONS:												
11		OUTER DIAMETER:		Outer Diar		(mm)							
12		INNER DIAMETER:		Inner Diam		(mm)							
13	SLOT DIMENSIONS:												
14		HS0		Hs0		(mm)							
15		HS1		Hs1		(mm)							
16		HS2		Hs2		(mm)							
17		BS0		Bs0		(mm)							
18		BS1		Bs1		(mm)							
19		BS2		Bs2		(mm)							
20		RS		Rs		(mm)							
21	TEETH DIMENSIONS:												
22		TOP:		Top Tooth		(mm)							
23		BOTTOM:		Bottom To		(mm)							
24	STATOR LENGTH:			Length of		(mm)							
25													
26	STATOR WINDING:												
27		WINDING TYPE:		Winding T									
28		COIL PITCH:		Average C									
29		BRANCHES:		Number of									
30		SLOT CONDUCTORS:		Number of									
31		STRANDS:		Number of									
32		WIRE DIAMETER:		Wire Diam		(mm)							
33		WIRE WRAP:		Wire Wrap		(mm)							
34		SLOT INSULATION:		Slot Insula		(mm)							
35		END ADJUSTMENT:		End Lengt		(mm)							
36		HALF-TURN LENGTH:		Coil Half-T		(mm)							
37													
38		FILL FACTOR:		Stator Slot		(%)							
39													

Set Boundary for Data Imported into Worksheet for RMxprt

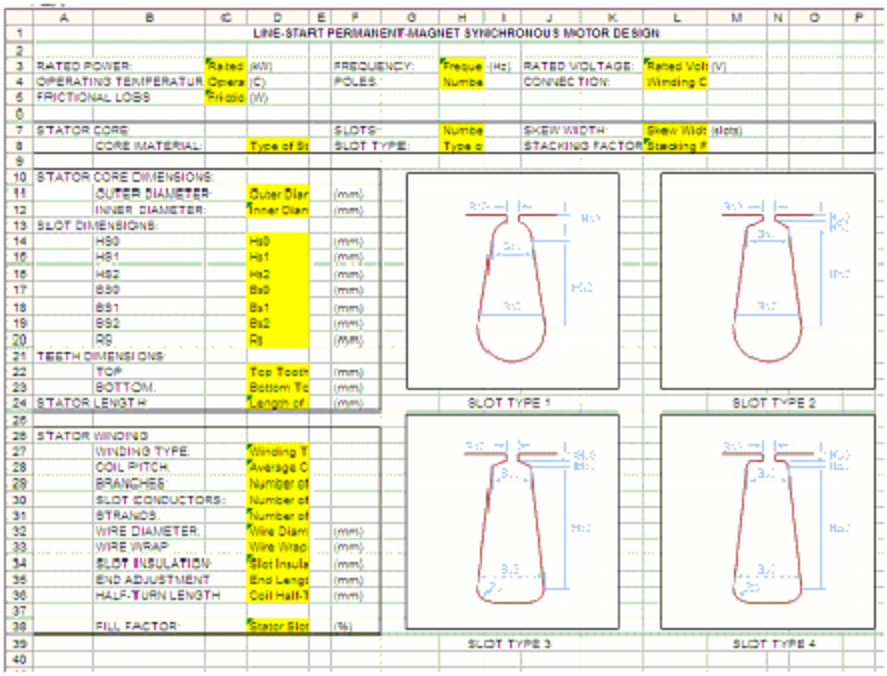
RMxprt automatically searches for matching key words while importing data into the **Microsoft Excel** worksheet. To reduce the searching space, and therefore the searching time, it is possible to set the searching boundary. **RMxprt** defines = "" as identification of boundary. In the following figure, for instance, the cells in the green-colored area are all keyed in with = "" to form boundary. While importing data, **RMxprt** will not search below or to the right of the boundary for matching key words.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	LINE-START PERMANENT-MAGNET SYNCHRONOUS MOTOR DESIGN													
2														
3	RATED POWER:		Rated	(kW)		FREQUENCY:		Freque	(Hz)	RATED VOLTAGE:		Rated Volt	(V)	
4	OPERATING TEMPERATUR		Opera	(C)		POLES:		Numbe		CONNECTION:		Winding C		
5	FRICTIONAL LOSS:		Frictio	(W)										
6														
7	STATOR CORE:					SLOTS:		Numbe		SKEW WIDTH:		Skew Widt	(slots)	
8		CORE MATERIAL:		Type of St		SLOT TYPE:		Type o		STACKING FACTOR		Stacking F		
9														
10	STATOR CORE DIMENSIONS:													
11		OUTER DIAMETER:		Outer Diar		(mm)								
12		INNER DIAMETER:		Inner Diam		(mm)								
13	SLOT DIMENSIONS:													
14		HS0		Hs0		(mm)								
15		HS1		Hs1		(mm)								
16		HS2		Hs2		(mm)								
17		BS0		Bs0		(mm)								
18		BS1		Bs1		(mm)								
19		BS2		Bs2		(mm)								
20		RS		Rs		(mm)								
21	TEETH DIMENSIONS:													
22		TOP:		Top Tooth		(mm)								
23		BOTTOM:		Bottom To		(mm)								
24	STATOR LENGTH:			Length of		(mm)								
25														
26	STATOR WINDING:													
27		WINDING TYPE:		Winding T										
28		COIL PITCH:		Average C										
29		BRANCHES:		Number of										
30		SLOT CONDUCTORS:		Number of										
31		STRANDS:		Number of										
32		WIRE DIAMETER:		Wire Diam		(mm)								
33		WIRE WRAP:		Wire Wrap		(mm)								
34		SLOT INSULATION:		Slot Insula		(mm)								
35		END ADJUSTMENT:		End Lengt		(mm)								
36		HALF-TURN LENGTH:		Coil Half-T		(mm)								
37														
38		FILL FACTOR:		Stator Slot		(%)								
39														

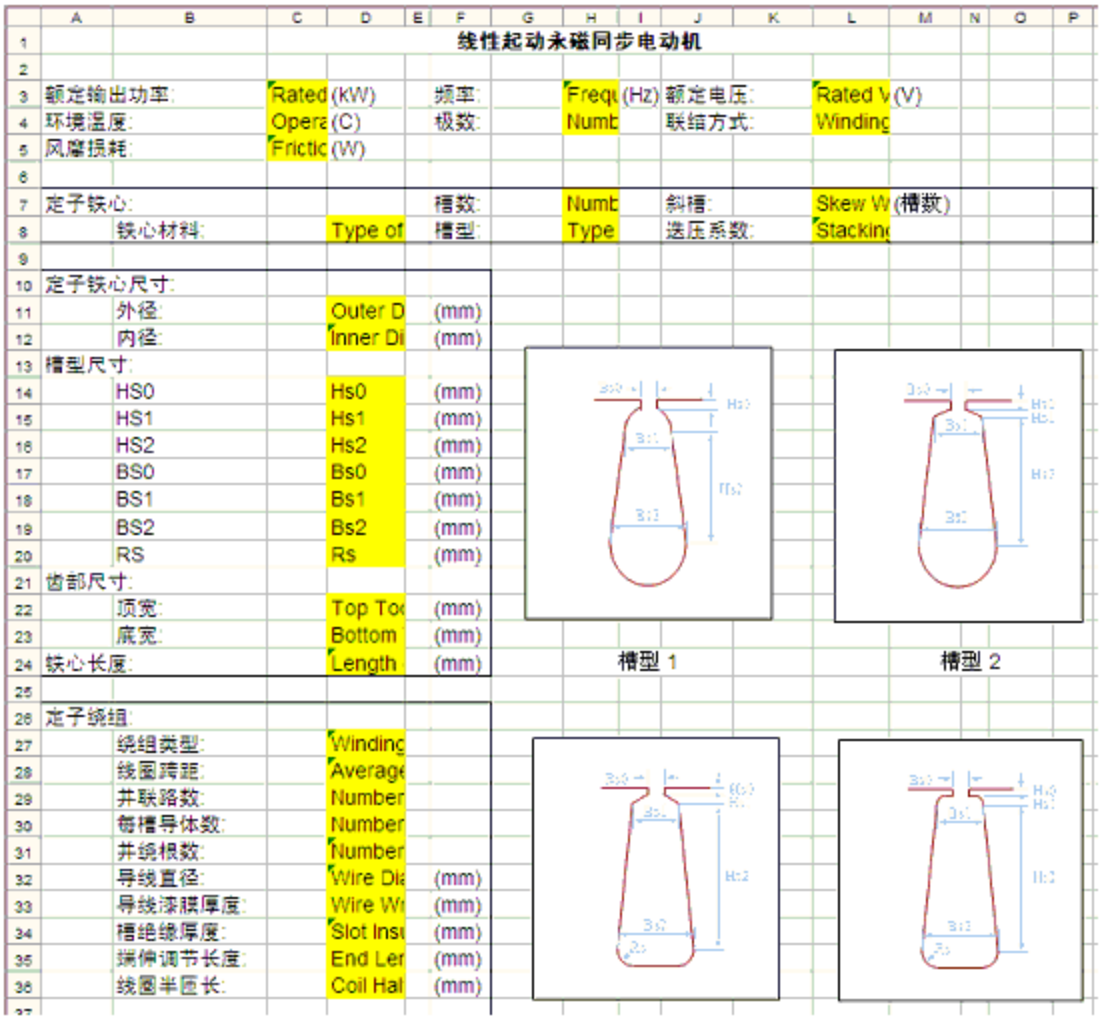
Insert Figures into Template for RMxprt

With resort to function **Paste**, it is possible to insert desired figures into the worksheet template. In the following example, for instance, four figures of slot types are inserted.



Use Different Languages for RMxprt Design Sheets

It is possible to use a preferred language other than English in the **Microsoft Excel** worksheet template. In the following example, for instance, Simplified Chinese is used.



Note	Key words are not allowed to be expressed in other languages.

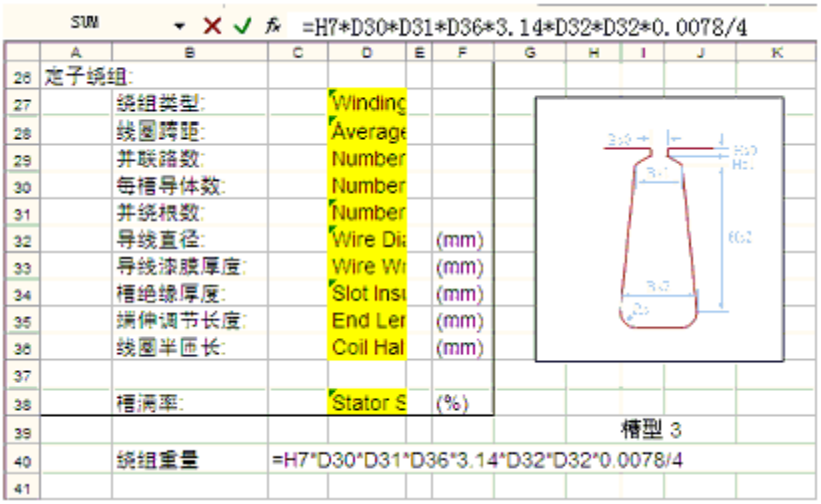
Post-process Data for RMxprt

Using the relevant functions of **Microsoft Excel**, it is possible to post-process data in the worksheet template. For example, calculate the weight of a winding using the following formula:

$$\begin{aligned} winding weight = {} & number\ of\ slots \times number\ of\ conductors\ per\ slot \\ & \times number\ of\ parallel\ wires \times length\ of\ half\ turns\ of\ coil \\ & \times sectional\ area\ of\ wire \times specific\ weight\ of\ wire \end{aligned}$$

In the following figure, the formula is entered into the relevant cell as:

=H7*D30*D31*D36*3.14*D32*D32*0.0078/4



Running an RMxprt Simulation

After you specify how RMxprt will compute the solution, begin the solution process.

1. Select a solution setup in the project tree.
2. Click **RMxprt>Analyze**.

To run more than one analysis at a time, follow the same procedure while a simulation is running. The next solution setup will be solved when the previous solution is complete.

To solve *every solution setup in a design*:

1. In the project tree, under the design you want to solve, select **Analysis**.
2. Click **RMxprt>Analyze All.**

   Each solution setup is solved in the order it appears in the project tree.

Aborting RMxprt Analyses

To end the solution process before it is complete:

- Right-click in the **Progress** window, and select **Abort** from the shortcut menu.

  The analysis stops immediately.

If you aborted the solution in the middle of an adaptive pass, the data for that pass or current frequency point is deleted. Any solutions that were completed prior to the one that was aborted are still available.

Re-solving an RMxprt Problem

If you modify a design after generating a solution, the solution in memory will no longer match the design. The solution setup with the invalid solution is marked with an *X* in the project tree and in the **Results** window.

To generate a new solution after modifying a design, follow the procedure for running a simulation:

1. Select a solution setup in the project tree.
2. Click **RMxprt>Analyze**.

Post Processing and Generating Reports in RMxprt

When RMxprt has completed a solution, you can display and analyze the results in the following ways:

- View solution data.
- Specify output variables.
- Export a model to be used in Maxwell 2D, Maxwell 3D, Twin Builder, or Motor-CAD™.
- Create a Customized Design Sheet
- Create Reports

Viewing RMxprt Solution Data

To access the **Solutions** dialog box, do one of the following:

- Click **RMxprt**>**Results>Solution Data**.
- Right-click **Results** in the project tree, and then click **Solution Data** on the shortcut menu.
- Right-click **Setup1** in the Project tree, and then click **Performance** on the shortcut menu.
- Click the **Solution Data** icon on the **Results** ribbon.

The **Solutions** dialog contains three tabs:

- **Performance** – this contains a Data field with a drop-down menu that allows you to view many different data tables, which vary with the machine type. Some examples are:
  - Aux Winding
  - Full Load Operation
  - Material Consumption
  - No Load Operation
  - Permanent Magnet
  - Rotor Data
  - Rated Parameters
  - Stator Slot
  - Stator Winding
  - Steady State Parameters
- **Design Sheet** – this displays the contexts of the **.res** file for the current setup. The file contains tables with information for such things as (depending on the machine type):
  - General Data
  - Stator Data
  - Rotor Data
  - Permanent Magnet Data
  - Material Consumption
  - Rated Operation
  - No-Load Operation
  - Steady State Parameters
  - No Load Magnetic Data
  - Full Load Data

- Winding Arrangement
- Transient FEA Input Data

To print the design sheet:

1. Right-click the design sheet to display the context menu.
2. Select **Print** from the shortcut menu.

   The **Print** dialog box appears.

3. Select the printer, and click **OK** to print.

- **Curves** – this displays the plots that were automatically generated by the solver. You can select these from a drop-down menu (menu contents vary with the machine type):
  - Input DC Current vs Speed
  - Efficiency vs Speed
  - Output Power vs Speed
  - Output Torque vs Speed
  - Cogging Torque in Two Teeth
  - Induced Coil Voltage at Rated Speed
  - Air Gap Flux Density
  - Induced Winding Phase Voltage at Rated Speed
  - Winding Currents Under Load
  - Phase Voltage Under Load

Note	You can also open the **Solution Data** dialog box directly by clicking the **Solution Data** icon on the Results ribbon.

  To print plots from the **Curve** tab:

1. Right-click on the desired plot to display the context menu.
2. Click **Print** on the menu to display the **Print** dialog.
3. Select the printer, and click **OK** to print.

Note	The context menu also provides commands that allow the user to change various plot characteristics such as: trace type and properties such as color; add/edit data markers and labels; edit axis, legend, and title properties. The plot image can also be copied to the clipboard for pasting in another application. Plot data can also be exported to various formatted text files that can then be imported into spreadsheets, etc. Refer to *Modifying the Background Properties of a Report* for additional information.

Browse Solutions in RMxprt

If you have run different solutions on a design, you can use the **RMxprt>Results>Browse Solutions** to look through and manage them.

1. Click **RMxprt>Results>Browse Solutions**

This displays the **Solutions** dialog with the **Browse** tab selected.

From here you can select a design, and view the Setup, Solution and State tables.

2. Click the **Properties** button to view the **Solution Browser** properties dialog box.

   This contains radio buttons for you to select the tree view. It can be organized as:

   - **Variation / Setup / Solution**
   - **Setup / Solution / Variation**
   - **Setup /Variation / Solution**

   Click **OK** to accept your selection and close the dialog.

3. From the **Solutions** dialog box, you can also select and delete solutions.

Creating a Maxwell Design from RMxprt

RMxprt provides a way to export solved models as either a Maxwell 2D or Maxwell 3D design.

- All the RMxprt machine types are supported.
- Setups for boundaries, excitations, parameters, etc. are automatically created.
- Variable definitions and assignments are automatically mapped from the solved RMxprt design to the Maxwell design.

Note	Creating a Maxwell2D/3D design from an old version (version 15 or older) of a solved RMxprt design results in the Maxwell2D/3D design being created *without* any variables even if the original RMxprt design uses variables. In such cases, the **Message Manager** displays a warning message informing the user that the Maxwell design being created does not contain any variable assignments from the RMxprt design; and recommends re-solving the setup to get the variable assignments.

To export the model to a Maxwell 2D Design or Maxwell 3D Design:

1. Click **RMxprt>Analysis Setup>Create Maxwell Design**

   The **Create Maxwell Design** dialog box appears.

2. Select one of the following from the **Type** drop-down list:
   - **Maxwell 2D Design**
   - **Maxwell 3D Design**
3. Select the setup you want to export from the **Solution Setup** drop-down list.
4. Optionally, click the **[...]** button to specify a **Variation** in the **Set Design Variation** dialog box. This allows the user to choose the default variable values to be used in the Maxwell design being created with variables.
5. Click **OK** to create the Maxwell design.

You can now work with the model in the Maxwell2D or Maxwell3D design, add boundaries and excitations, perform analyses, and analyze results.

Note	• For Maxwell 2D/3D designs for motor applications, the computed output mechanical power at a given constant speed may not reach the desired rated power. Therefore, it is useful to apply constant power as the mechanical load. In

	Maxwell 2D/3D designs created by RMxprt, the mechanical transient is setup according to the rated mechanical output power, and equivalent damping is added to accelerate the process approaching to the rated output power. This setup is visible when **Consider Mechanical Transient** on the Mechanical tab of the Motion Setup dialog box is checked. • For Maxwell 2D/3D designs of AC machines, RMxprt automatically enables an alternating flux (AF) model in the voltage definition for each winding to reduce transient solution time to reach steady-state when the Maxwell 2D or 3D model is created. • For Maxwell 2D designs of double-cage induction machines with separate end rings, and of synchronous machines with separate dampers for each pole, RMxprt automatically creates multiple end connection excitations in the Maxwell 2D design. For synchronous machines, if the created geometry model has only one pole, it is changed to two poles, with two separate end connection excitations. This enables simulation of double-cage induction machines with separate rings, as well as synchronous machines with separate dampers for each pole.

Related Topics

Defining Maxwell 3D Boundary Conditions

Defining Maxwell 3D Excitations

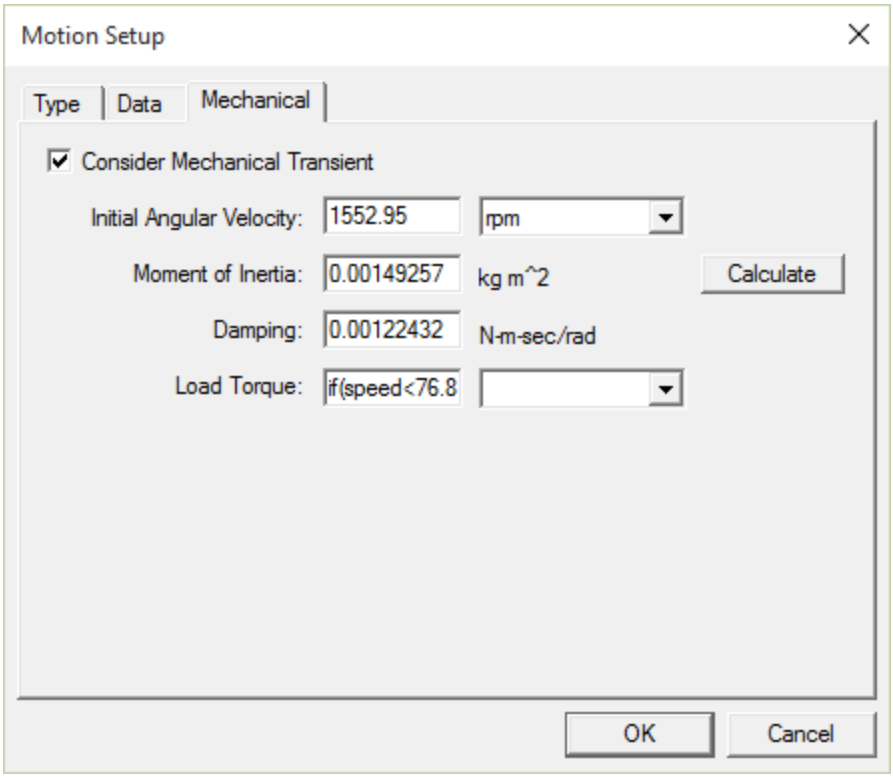
Defining Maxwell 2D Boundary Conditions

Defining Maxwell 2D Excitations

Speed Adjustment to Achieve to Rated Output Power

Speed Adjustment to Achieve to Rated Output Power

The speed setup in all RMxprt created Maxwell designs is the estimated rated speed which is computed based on the rated output power using Equivalent Magnetic Circuit (EMC) method in RMxprt. The EMC method may not be very accurate in some cases, especially when some part of the magnetic circuit is highly saturated. In order to obtain the correct rated performance in Maxwell designs, the speed setup by RMxprt based on the estimated rated speed should be adjusted so that the simulated mechanical output power matches the specified rated output power. *All rated performance data, such as torque, speed and current, must be obtained at the rated output power!* For this purpose, all RMxprt created Maxwell designs have an initial mechanical transient setup to automatically adjust the speed to approach the rated mechanical output power. This initial transient mechanical setup will be lost if you close the project without checking the **Consider Mechanical Transient** check box as shown in the following example.



However, using the mechanical transient to reach to the rated output power may need much longer simulation time especially when the rotor Moment of Inertia is large. Reducing the value of the Moment of Inertia may cause speed oscillation.

The following two methods may be used to adjust the speed to achieve to the rated output power:

- Using the initial mechanical transient setup by RMxprt
- Adjusting speed manually for a constant speed setup

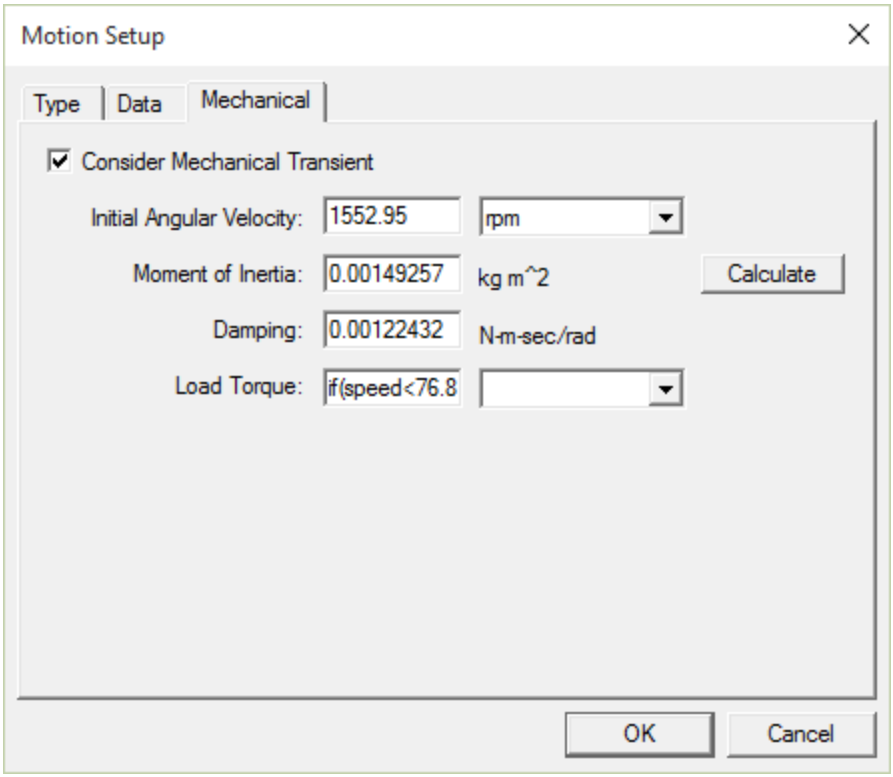
Related Topics

Using the Initial Mechanical Transient Setup by RMxprt

Adjusting Speed Manually for a Constant Speed Setup

Using the Initial Mechanical Transient Setup by RMxprt

To use this method, you need to check **Consider Mechanical Transient** in the **Mechanical** tab of the **Motion Setup** dialog box immediately after the design is created from RMxprt. An example showing the initial values for mechanical transient setup is shown below.



Note that if you close the design without checking it, and open the design later, this initial mechanical transient setup will loss.

The method for using the mechanical transient setup is as follows:

In the Mechanical tab of the Motion Setup panel, when **Consider Mechanical Transient** is checked, the displayed **Load Torque** is actually the "Driving Torque" because the positive torque is defined in the positive speed direction. Therefore, the real positive Load Torque must be defined with a negative value. In order to automatically adjust the speed for the motor to generate the specified output power, you need to set the load torque curve with Constant Power, that is:

Tload = – Pspec / speed

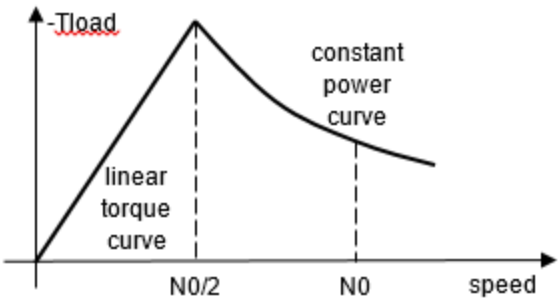
where **speed** is the real rotor speed, and **Pspec** is a constant value.

In order to apply this load torque curve for any initial speed, the load torque curve is defined as:

Tload = if(speed<N0/2, -4*Pspec/N0/N0*speed, -Pspec/speed)

where **N0** is the *synchronous speed* for synchronous motors, or the *rated speed* for other motor types in (rad/s).

The load torque is shown below, which is applicable for all motor types excluding synchronous motors.

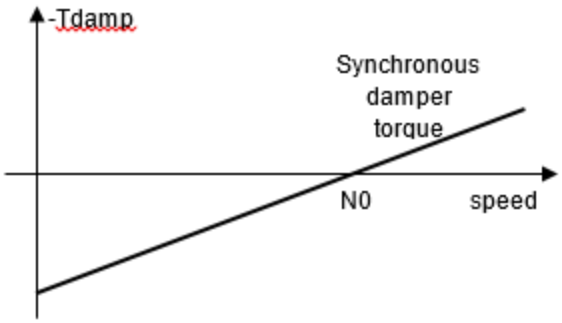


For synchronous motors, the motor driving torque varies with torque angle which is the difference between the phase angles of the applied and the induced voltages. If the initial speed is set as the synchronous speed, and the initial torque angle is set smaller than the real value, the motor driving torque will be smaller than the specified load torque. With mechanical transient, the speed will decrease, which makes the torque angle and driving torque increase. When the driving torque equals the load torque, the speed, which is lower than the synchronous speed, will stay constant temporarily. Since the speed is lower than the synchronous speed, the torque angle continues to increase, which makes the driving torque greater than the load torque, and therefore the speed increases. When the speed reaches the synchronous speed, the torque angle will be too large, which will increase the speed to exceed the synchronous speed. Therefore, the speed will oscillate about the synchronous speed. This oscillation will decay to zero if the synchronous motor has a damper cage winding. The decaying time depends on the strength of the damping effects.

In order to speed up the decaying process, an additional load torque component to model the damping effects around the synchronous speed is set for all types of synchronous motors, as expressed below:

Tdamp = - Tst * (speed/N0 - 1)

where **Tst** is set as the rated torque. You may increase **Tst** to increase the damping effects. The damping torque curve is shown below.

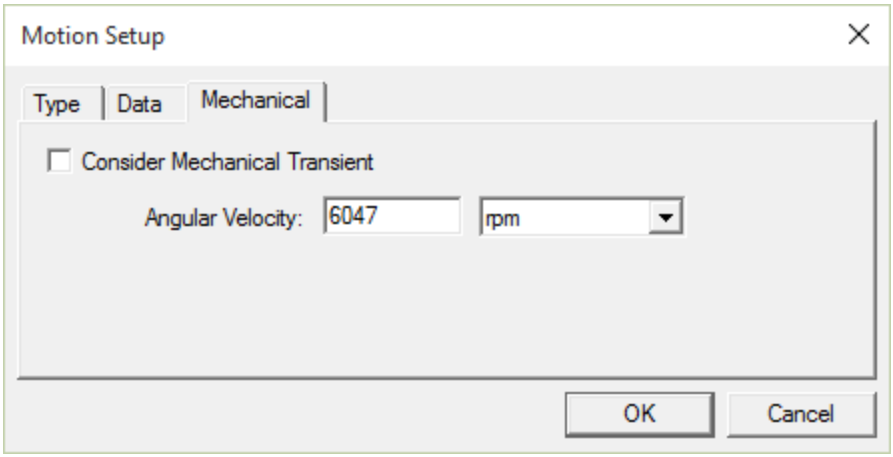


Related Topics

Speed Adjustment to Achieve to Rated Output Power

Adjusting Speed Manually for a Constant Speed Setup

The specified rated output power can be reached by manually adjusting the input speed without checking **Consider Mechanical Transient**.



This adjustment method may require iteration, as described below:

Determine the solution at the RMxprt setting speed.

Using a Permanent Magnet DC motor as an example, assume that the solution data for the Maxwell 2D design originally created by RMxprt are:

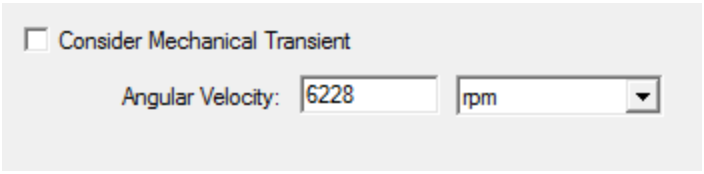
- The period is 4.96 ms based on the original setting speed n0 = 6047rpm for a 4-pole motor.
- The average torque in the last period, from 5.04 ms to 10 ms, is T0 = 1.4044 Nm
- The simulated output power is 889.3 W.

If the specified rated output power is 200 W, in order to reduce the motor driving output power, we need to increase the motor speed.

For the next iteration, try increasing the speed by 3%, that is: n1 = n0 * (1+3%) = 6228 rpm.

First Iteration – Determine the solution at the first adjusted speed.

When the speed is set to be n1 = 6228 rpm:



- The period for this speed is 4.82 ms.
- The new simulated average torque in the last period is T1 = 0.7276 Nm
- The output power is P1 = 0.7276 * 6228 * 2*pi/60 = 474.5 W.

If the output power of 474.5 W is acceptable, you can stop here. Otherwise, you need to continue to adjust the speed as described below.

Second Iteration – Determine the solution at the second adjusted speed.

Now, we have two solutions of (n0, T0) = (6047 rpm, 1.4044 Nm) and (n1, T1) = (6228 rpm, 0.7276 Nm). Assuming the motor driving torque **T** at any speed **n** can be obtained from linear interpolation and extrapolation, the motor driving torque can be expressed as:

T = T0 + (T1-T0) / (n1-n0) * (n-n0) = 1.4044 – 0.003739 * (n – 6047) = 24.0155 – 0.003739 * n

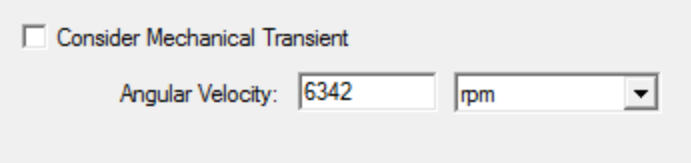
In order to obtain the specified rated output power, the load torque can expressed as:

T = 200 / n * 60/(2 pi) = 1910 / n

The new adjusted speed **n2** can be obtained by solving the above two equations, the resulting value being:

n2 = 6342 rpm

Thus, if we set the speed to be n2 = 6342 rpm.



- The period for this speed is 4.73 ms.
- The new simulated average torque in the last period is T2 = 0.3065 Nm.
- Now, the output power is: P2 = 0.3065 * 6342 * 2*pi/60 = 203.6 W, which is very close to the specified rated output power of 200 W.

Comparing the discrepancy between RMxprt and Maxwell 2D results after this second iteration:

	RMxprt	Maxwell 2D	Error
Speed	6047 rpm	6342 rpm	-4.65%
Torque	0.3216 Nm	0.3065 Nm	4.93%
Output Power	200 W	203.6 W	-1.77%

The above table shows that for a small change of speed in Maxwell setup, the torque change may be significant. In the example, the Maxwell 2D result of torque at n0 = 6047 rpm is T0 = 1.4044 Nm, while the torque at n2 = 6342 rpm is T2 = 0.3065Nm. The speed change of 4.65% results in a torque change of 358%.

In the above method, the algorithm for the speed adjustment is suitable for most motor types including (PM)DC motors, BLDC motors, universal motors, induction motors, and switched reluctance motors. For induction motors, the first adjusted speed can be obtained by increasing or decreasing the speed slip with certain percentage.

Note	• The speed setup by RMxprt is just an initial setup. Users must adjust the speed if the output power is far away from the specified rated output power.

	• This method is suitable for all (PM)DC motors, BLDC motors, universal motors, induction motors, and switched reluctance motors. For all synchronous machines, users must adjust the power angle or the torque angle if the output power is far away from the rated output power.

Related Topics

Speed Adjustment to Achieve to Rated Output Power

Exporting a Simplorer Model, MotorCAD Project, or Customized Design Sheet

To export the model to a Simplorer model, a MotorCAD™ project, or Customized Design Sheet:

1. First, solve the design for the specific solution setup from which the export is desired.

Note	Attempting to export without the requisite files present automatically launches simulation to generate them.

2. Click **RMxprt>Analysis Setup>Export** to open the **Export** dialog box.

   You can also right-click on **Analysis** or **Analysis>Setup** in the Project Manager and select **Export**.

3. Select one of the following from the Type drop-down list:
   - **Simplorer Model**
   - **Customized Design Sheet**
   - **MotorCAD Project**
4. Select the setup you want to export from the **Solution Setup** drop-down list.
5. Optionally, click the Variation ellipsis [**...**] button to specify a design **Variation to use during export**.
6. Specify the **Path** to store the exported files in by clicking the Path ellipsis [...] button and browsing to the desired directory. You can also create a new directory, if desired.
7. Click **OK**.

   – For **MotorCAD Project**, a <*project_name*>**.mot** file that includes all machine and housing data will be generated. You can then open the project in Motor-CAD for analysis.

   – For **Customized Design Sheets**, a <*design_sheet_name*>**.xls** file will be generated.

   – For **Simplorer** models, three files will be exported:
   - <*project name*>_<*design name*>_<*setup name*>.sml
   - <*project name*>_<*design name*>_<*setup name*>_*signals*.sml
   - <*project name*>_<*design name*>_<*setup name*>_*SimCkt*.vbs

   You can now use the Twin Builder **Tools>Run Script** command to run the .vbs script to generate the components on the Twin Builder schematic. (Refer to the Twin Builder help for detailed information on running scripts.)

You can now work with the model in the product to which you exported it, and perform other kinds of analyses.

Related Topics

Generating a Custom Design Sheet for RMxprt

Creating an Icepak Design

RMxprt provides a way to export solved models as Icepak designs.

To export the model as an Icepak Design:

1. Click **RMxprt>Analysis Setup>Create Icepak Design**; or right-click on the analysis setup in the Project tree and select **Create Icepak Design** from the context menu.

   The **Create Icepak Design** dialog box appears.
2. Select the setup you want to export from the **Solution Setup** drop-down list.
3. Optionally, click the **[...]** button to specify a **Variation** in the **Set Design Variation** dialog box. This allows the user to choose either the nominal design or to choose the variation to be used in the Icepak design being created.
4. Enter or choose the **Path** where the Icepak project is to be output.
5. Do one of the following:
   a. Click **OK** without selecting **Launch Icepak application** to generate the Icepak project and put it in the designated folder.
   b. Select **Launch Icepak application** and click **OK** to generate the Icepak project in the designated folder, and automatically open the Icepak design in the Icepak application.

Note	• If you want to launch the Icepak application, ensure that the Workbench path is set in the Tools/Options/General Option/Miscellaneous tab. • Refer to the Icepak documentation for information on working with Icepak designs.

Creating Reports in RMxprt

After RMxprt has generated a solution, all of the results for that solution are available for analysis. One of the ways you can analyze your solution data is to create a report, or graphical representation, that displays the relationship between a design's values and the corresponding analysis results. Reports are created using the **RMxprt>Results>Create RMxprt Report>**<*report_type*> command. The available options in the resulting **Reports** dialog box depend on the report type you create and the available solution data. You can also create reports using the **RMxprt>Results>Create Quick Report** command. The **Quick Report** feature lets you select from a list of predefined categories (such as coil voltage or torque) from which to create a rectangular plot.

If you have created custom report templates (for example, including your company name or other format changes), you can also create a report based on that template by selecting **RMxprt>Results>Report Templates>***<templateName>*.

You can also Create a Report from a Report Data (`.rdat`) file by selecting **RMxprt>Results>Create Report from File**, then browse to the desired `.rdat` file and open it. (Refer to *Performance Computation Based on FEA Created ROM for Induction Machines for an example.*)

The report appears in the view window and is listed in the project tree.

Related Topics

Selecting the Display Type in RMxprt

Modifying Reports in RMxprt

To modify the data that is plotted in a report:

1. In the project tree, right-click the report you want to modify.
2. Select **Modify Report** from the shortcut menu.
3. Modify the selections in the Report dialog box as needed.
4. Click **Close** when you are finished modifying the report.

   The updated report appears in the view window.

To update all modified reports:

Click **RMxprt>Results>Update Reports**.

Opening All Reports in RMxprt

To open all reports for a project:

Click RMxprt>Results>Open All Reports.

This opens all reports.

To simplify viewing and comparisons, it may be helpful to use **Window>Cascade** or **Window>Tile Horizontally** or **Window>Tile Vertically** commands.

To close all open reports:

Click **Window>Close All**.

Deleting All Reports in RMxprt

To delete all reports for a project:

Click **RMxprt>Results>Delete All Reports**.

This deletes all reports for the project.

Selecting the Display Type in RMxprt

The information in a report can be displayed in several formats. Select from the following **Display Type** formats in the **Create Report** dialog box:

Rectangular Plot	A 2D rectangular (x-y) graph.
Rectangular Stacked Plot	A series of 2D rectangular (x-y) graphs stacked vertically, sharing a common x-axis scale, but with each trace having its own y-axis scale.
3D Rectangular Plot	A 3D rectangular (x-y-z) graph.
3D Rectangular Bar Plot	A 3D rectangular (x-y-z) bar graph.
3D Polar Plot	A 3D circular plot divided by spherical coordinates.
Data Table	A grid with rows and columns that displays, in numeric form, selected quantities against a swept variable or another quantity.
Rectangular Contour Plot	A rectangular (x-y-z) graph. Contour plots are useful to visualize surfaces (for e.g. Directivity as a function of phi/theta).

Creating 2D Rectangular Plots in RMxprt

A rectangular plot is a 2D, x-y graph of results.

1. Click **RMxprt>Results>Create RMxprt Report>Rectangular Plot**.

   The **Report** dialog appears.
2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   a. Solution field with a drop down selection list. This lists the available solutions, whether sweeps or adaptive passes.
   b. Domain field with a pull down selection list containing options for plotting vs ElectricDegree, Speed, or a user-selected Parameter.
3. Under the **Trace** tab, **Y** component section, specify the information to plot along the y-axis:
   a. In the **Category** list, click the type of information to plot.
   b. In the **Quantity** list, click the value to plot.
   c. In the **Function** list, click the mathematical function of the quantity to plot.
   d. Value field displays the currently specified Quantity and Function. You can edit this field directly.

Note	Color shows valid expression.

e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.

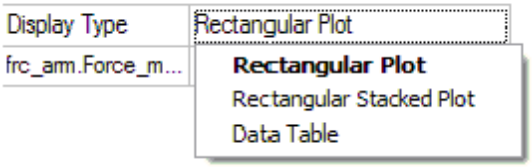
4. On the **Trace** tab, **X** (Primary sweep) line, specify the quantity to plot along the x-axis in one of the following ways:
   - Select the sweep variable to use from the drop down list.
   - If sweeps are available, you can select the browse [...] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time sweeps), or **Use all values** (the default setting). The quantity will be plotted against the primary sweep variable listed.
5. On the **Families** tab, confirm or modify the sweep variables that will be plotted.
6. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog box. The default name is based on the Report Category you selected, (for example, Force Plot *n* or Output Variables Plot *n*). You can edit the plot names in the project tree and the plot header text in the report synchronizes.

   The function of the selected quantity will be plotted against the swept variable values or quantities you specified on an x-y graph. The plot is listed under **Results** in the project tree and the traces are listed under the plot. When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.

10. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

You can also modify the display type of an existing plot from the Properties dialog for that plot. Select the Report icon in the Project tree to display the Properties dialog box. Selecting the Display Type field displays a menu with selections available for that plot.



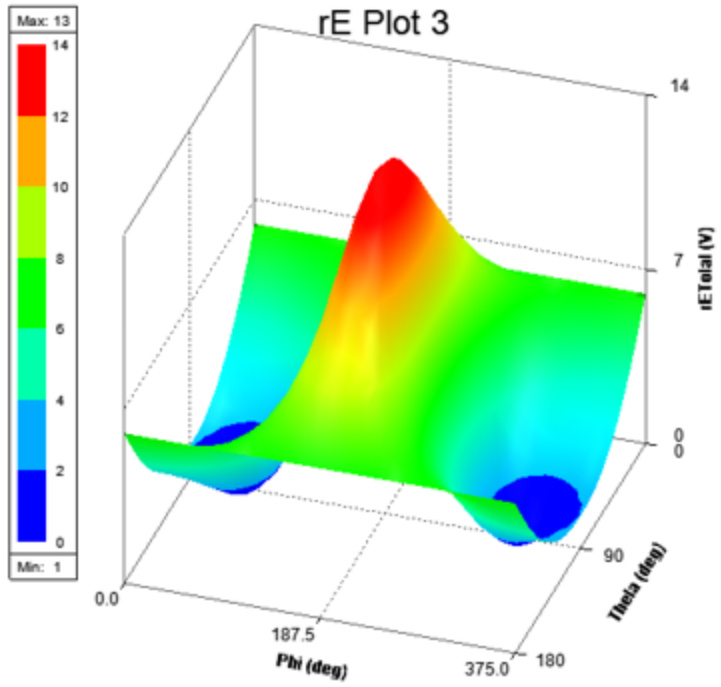
Related Topics

Sweeping a Variable

Working with Traces

Creating 3D Rectangular Plots in RMxprt

Below is an example of a 3D rectangular plot.



Working with a 3D Rectangular Plot

You can Rotate, Zoom and Pan a plot. When you rotate, the Cartesian grid responds so that the curve always remains in front and the grids behind.

Clicking on a plot entity selects it, highlighting the selected entity in bold.

Double-clicking anywhere in the plot brings up the Properties dialog box. the properties are grouped appropriately under various tabs, which correspond to plot entities:

- General: For general plot properties such as Visual Detail level and background color
- Header: Properties related to plot Header/Title.
- Axis [X|Y|Z]: Properties related to the 3 axes
- Grid [XY|YZ|ZX]: Properties related to the 3 grids
- ColorKey: Properties related to ColorKey, including borders, background, Min and Max, as well as number format and precision.
- Contour: Properties related to contouring of all curves/surfaces
- Surface: Properties related to the curve

Selecting a property also displays its properties in the Property window. You can edit the properties to customize the appearance of the plot. See "Controlling Visual Detail in a 3D Plot" on page 26-159.

Creating a 3D Rectangular Plot

1. Click **RMxprt>Results>Create RMxprt Report>3D Rectangular Plot** .
2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   a. Solution field with a drop down selection list. This lists the available solutions, whether sweeps or adaptive passes.
   b. Domain field with a pull down selection list containing options for plotting vs ElectricDegree, Speed, or a user-selected Parameter.
3. Under the **Z** tab, specify the information to plot along the z-axis:
   a. In the **Category** list, click the type of information to plot.
   b. In the **Quantity** list, click the value to plot.
   c. In the **Function** list, click the mathematical function of the quantity to plot.
   d. The **Value** field displays the currently specified Quantity and Function. You can edit this field directly.

Note	Color shows valid expression.

   e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.
4. On the **Trace** tab, **Y** (**Secondary Sweep**) lines, specify the information to plot along the y-axis in one of the following ways:
   - Select the sweep variable to use from the Secondary Sweep drop down list.
   - If sweeps are available, you can also select the browse [...] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time sweeps), or **Use all values** (the default setting). The quantity will be plotted against the primary sweep variable listed.
5. On the **Trace** tab, **X** (**Primary Sweep**) lines, specify the information to plot along the x-axis in one of the following ways:
   - Select the sweep variable to use from the Primary Sweep drop down list.
   - If sweeps are available, you can also select the browse [...] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time sweeps), or **Use all values** (the default setting). The quantity will be plotted against the primary sweep variable listed.
6. On the **Families** tab, confirm or modify the sweep variables that will be plotted.
7. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog box. The default name is based on the Report Category you selected, (for example, Force Plot *n* or Output Variables Plot *n*). You can edit the plot names in the project tree and the plot header text in the report synchronizes.

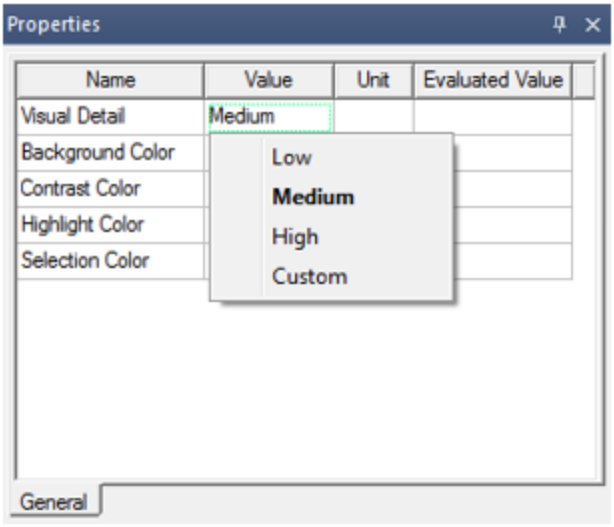
   The function of the selected quantity or quantities will be plotted against the values you specified on an x-y-z graph. The plot is listed under **Results** in the project tree.When you select the traces or plots, axis or grid labels, plot header, color key, or variable labels, their

properties are displayed in the Properties window. The properties for each plot element can be edited directly to modify the plot content and appearance. See Modifying the Background Properties of a Report.

8. Optionally, add another trace to the plot by following the procedure above, using **Add Trace rather than New Report**.

Controlling Visual Detail in a 3D Plot

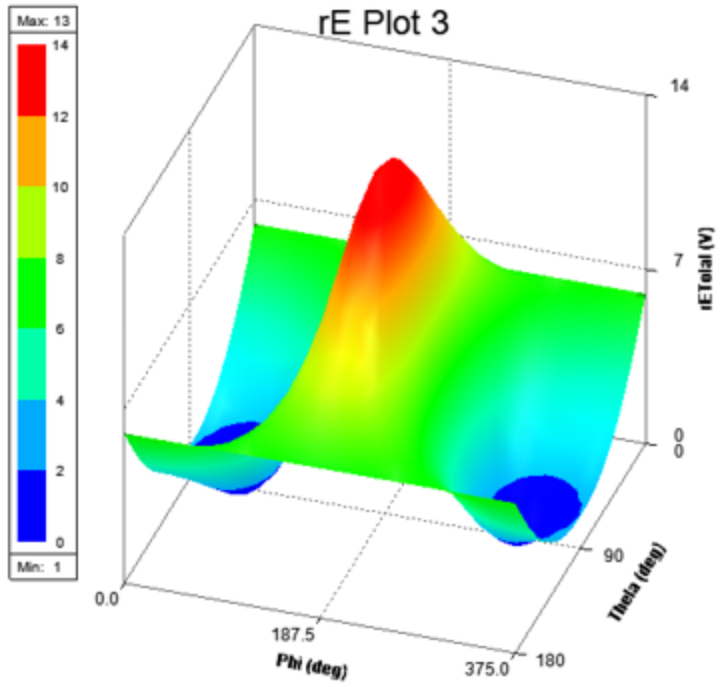
If a particular plot seems busy with information, you can edit plot properties, such as Axis and Grid Attributes for discrete levels of visual detail to improve readability. Double-click anywhere on a plot to display the Properties dialog box. The Visual Detail property on the General tab also provides control suited to different screen and plot sizes.



The Visual Detail menu has four options: Low, Medium (the default), High, and Custom. If you select any Visual Detail, the 3D plot is rendered according to the selected Visual Detail level and the properties reflect the values chosen for the selected visual detail level. From this predefined visual detail level, if you modify any properties, Visual Detail is automatically set to Custom (or to another predefined visual detail level if the edits happen to match the settings for that level).

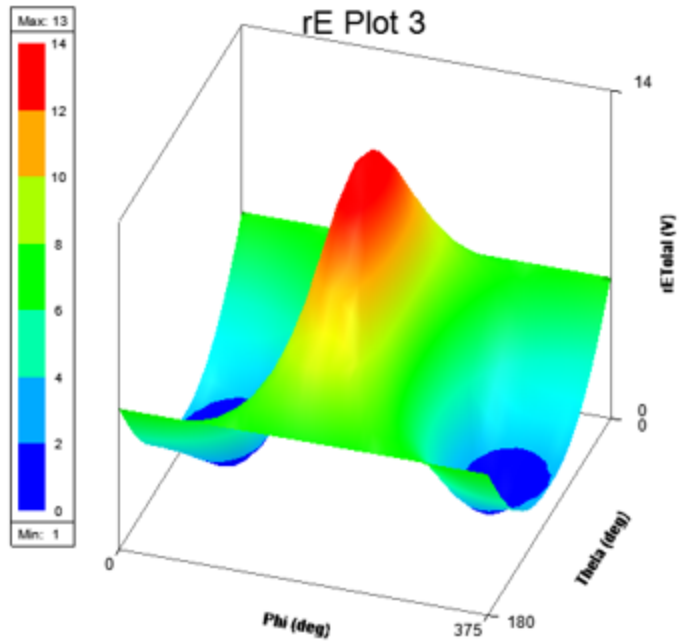
You can also manually set Visual Detail to Custom. In such a case, Custom will inherit property values corresponding to the previous level. This ensures that you can customize settings starting from a baseline provided by the preconfigured Low, Medium or High Visual Detail levels.

3D Rectangular Plot with Medium Visual Detail



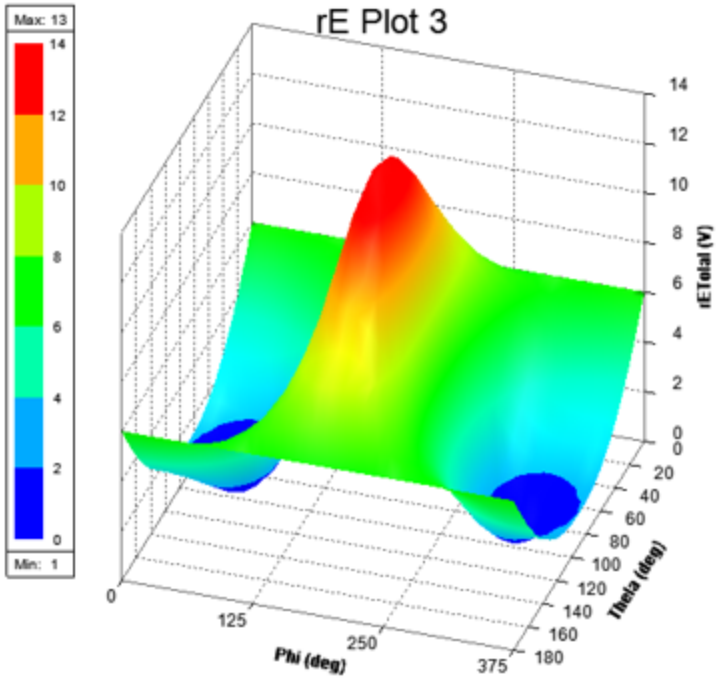
On creation, a 3D Rectangular Plot has Visual Detail set to Medium and looks and feels as shown above. Specifically, under Medium Visual Detail level, a 3D Rectangular Plot has 3 ticks per axis (X, Y, Z axis) which will show min, max and middle value. This setting also shows axes labels.

3D Rectangular Plot with Low Visual Detail



With Visual Detail set to Low, a 3D Rectangular Plot shows axes with 2 ticks corresponding to min and max values. It also shows axes labels and grid borders.

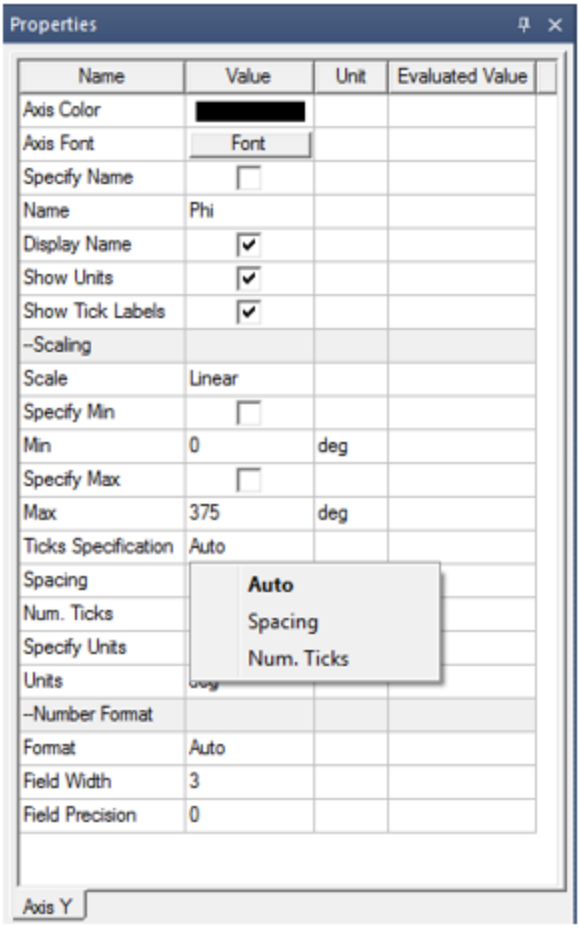
3D Rectangular Plot with High Visual Detail



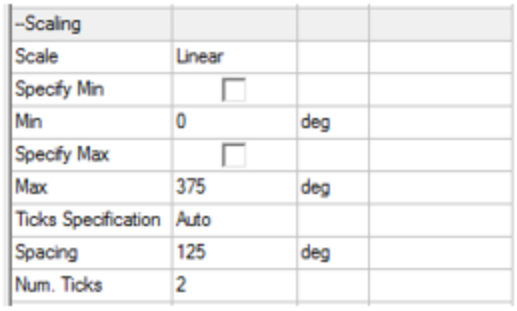
With Visual Detail set to High, a 3D Rectangular Plot shows all Cartesian axes and grids together with all ticks and axes labels.

Axis Properties: Ticks Specification and Num. Ticks

Ticks Specification is available on Axis properties, as shown below:



Ticks Specification is a menu with possible values as Auto, Spacing, and Num. Ticks, with Auto being the default value. If Ticks Specification is Auto, then a spacing value is automatically calculated and used to calculate and display the tick labels. **Spacing** shows the calculated value, and Num. Ticks shows the number of ticks based on this spacing value, as shown below:



You can edit the **Spacing** field when Ticks Specification is set to Spacing; otherwise, it is read only.

You can edit the **Num. Ticks** field when Ticks Specification is Num. Ticks; otherwise, it is read only.

Valid Num. Ticks are between 0 and 100, including 0 and 100. If you enter an invalid value, an error message is shown. If you enter a spacing value that results in number of ticks greater than 100, then an appropriate value is shown.

- If Num. Ticks is 0, then no ticks are shown on the axis.
- If Num. Ticks is 1, then only the max value tick is shown on the axis.
- If Num. Ticks is 2, then only the min and max value ticks are shown on the axis.
- If Num. Ticks is greater than 2, then evenly spaced ticks (including min and max) are shown on the axis.

Note	With the addition of the Ticks Specification property to Axis properties, the **Specify Spacing** property was removed as an Axis property. • If an R18.0 or R18.1 project is opened with **Specify Spacing** as Unchecked, Ticks Specification is set to Auto. • If an R18.0 or R18.1 project is opened with **Specify Spacing** as Checked, Ticks Specification is set to Spacing.

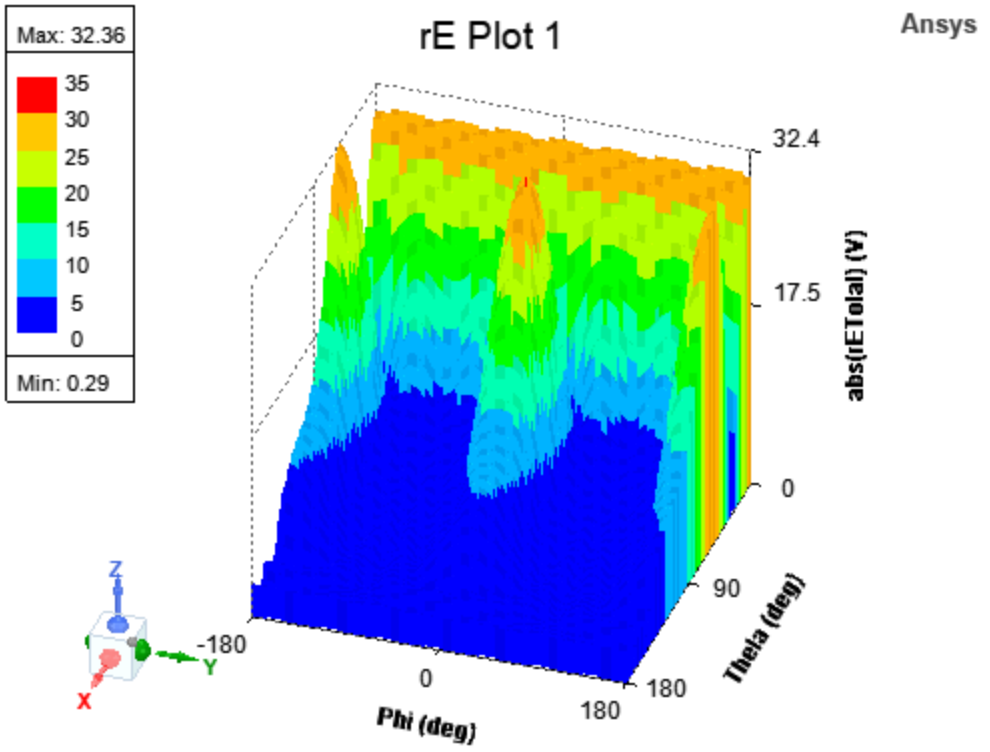
Related Topics

Sweeping a Variable

Working with Traces

Creating 3D Rectangular Bar Plots

This is a 3D, x-y-z graph of results as rectangular bars.

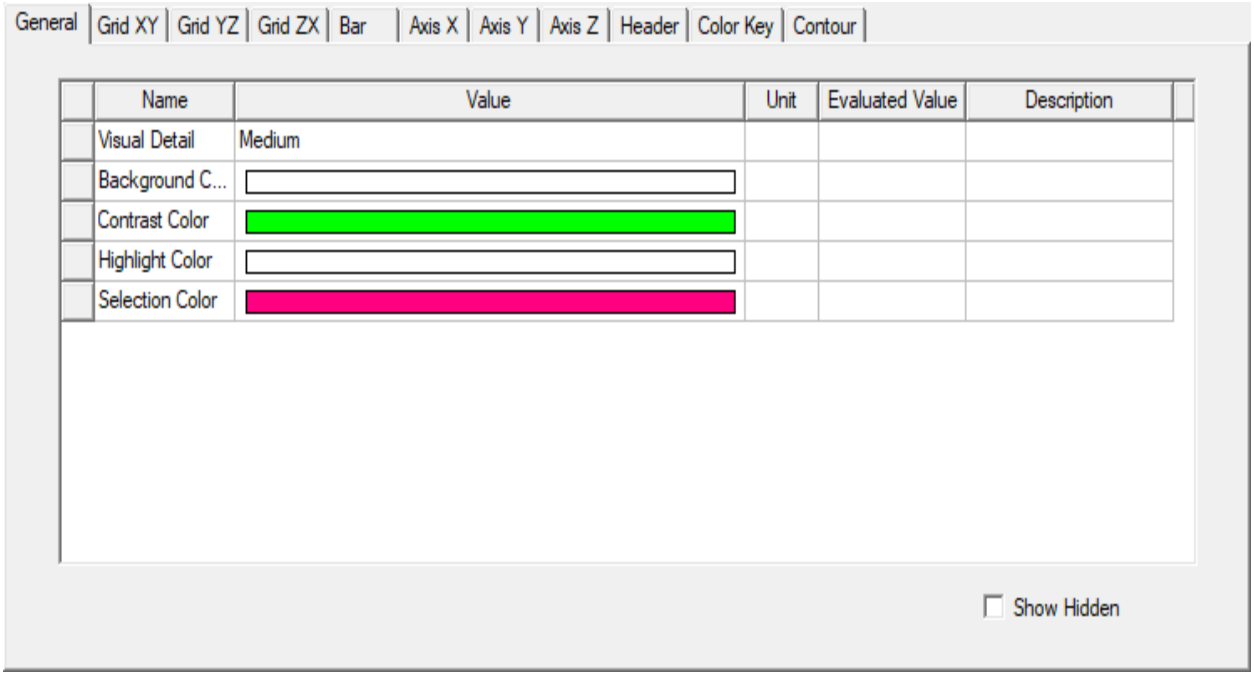


Working with a 3D Rectangular Plot

You can Rotate, Zoom, and Pan a plot. When you rotate, the Cartesian grid responds so that the curve always remains in front and the grids behind.

Clicking on a plot entity selects it, highlighting the selected entity in bold.

Double-clicking anywhere in the plot brings up the **Properties** dialog box.



The properties tabs for a 3D Rectangular Bar plot are:

- General: For general plot properties such as Visual Detail level and background color
- Grid [XY|YZ|ZX]: Properties related to the 3 grids
- Bar: Properties related to rectangular bars.
- Axis [X|Y|Z]: Properties related to the 3 axes
- Header: Properties related to plot Header/Title.
- ColorKey: Properties related to ColorKey, including borders, background, Min and Max, as well as number format and precision.
- Contour: Properties related to contouring of all surfaces

Selecting a property also displays its properties in the Property window. You can edit the properties to customize the appearance of the plot. See "Controlling Visual Detail in a 3D Rectangular Bar Plot" on page 26-169.

Creating a 3D Rectangular Bar Plot

1. On the **Results** menu (under the **Maxwell** menu or right-click **Results** in the Project Manager), click **Create *<type>* Report**, and select **3D Rectangular Bar plot** from the report type menu, or select the Results tab of the ribbon, and select the 3D Rectangular Bar icon for the Report type you want to create.

   The *Report* dialog box appears.
2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   a. Solution field with a drop-down selection list. This lists the available solutions, whether sweeps or adaptive passes.

b. Geometry field with a drop-down selection list. For field and radiated field reports, this applies the quantity to a geometry or radiated field setup.

3. Under the **Trace** tab **Z** Component area, specify the information to plot along the z-axis:
   a. In the **Category** list, click the type of information to plot. The category you select provides the default plot name.
   b. In the **Quantity** list, click the value to plot.
   c. In the **Function** list, click the mathematical function of the quantity to plot.
   d. The **Value** field displays the currently specified Quantity and Function. You can edit this field directly.

      > **Note:**
      >
      > Color shows valid expression.

   e. **Range Function** button -- opens the *Set Range Function* dialog box. This applies currently specified Quantity and Function.
4. On the **Trace** tab **Y** (Secondary sweep) lines, specify the information to plot along the y-axis in one of the following ways:
   - Select the sweep variable to use from the drop-down list.
   - If sweeps are available, you can select the browse button to display a dialog that lets you select particular values. The quantity will be plotted against the primary sweep variable listed.
5. On the **Trace** tab **X** (Primary sweep) lines, specify the information to plot along the x-axis in one of the following ways:
   - Select the sweep variable to use from the drop-down list.
   - If sweeps are available, you can select the browse button to display a dialog that lets you select particular values. The quantity will be plotted against the primary sweep variable listed.
6. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables **Add Trace** on the **Report** dialog box. The default name is based on the Report Category you selected, (for example, S Parameter Plot *n* or rE Plot *n*). You can edit the plot names in the project tree and the plot header text in the report synchronizes.

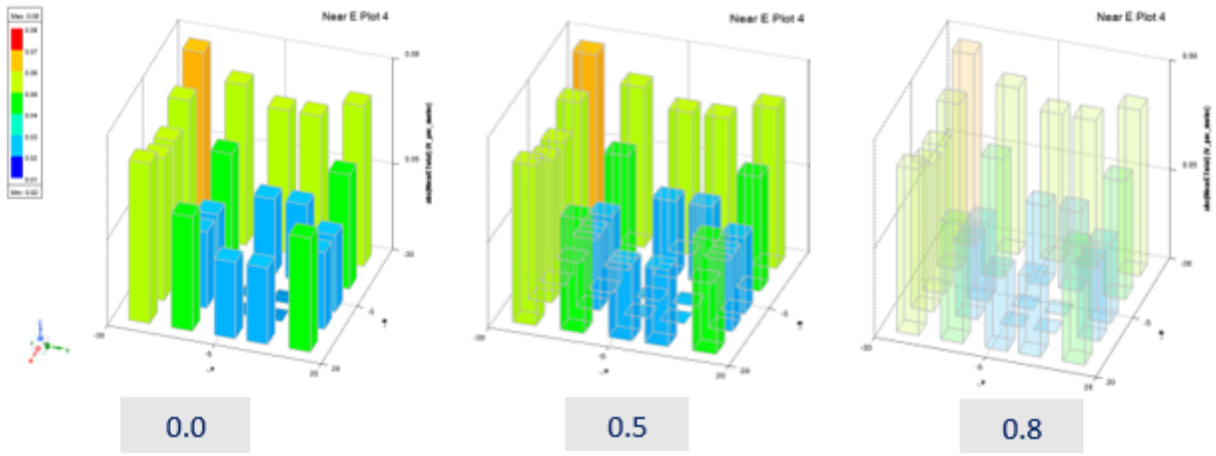
   The function of the selected quantity or quantities will be plotted against the values you specified on an x-y-z graph. The plot is listed under **Results** in the Project Manager. When you select the traces or plots, axis or grid labels, plot header, color key, or variable labels, their properties are displayed in the Properties window. The properties for each plot element can be edited directly to modify the plot content and appearance. See Modifying the Background Properties of a Report.
7. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

3D Rectangular Bar Customization

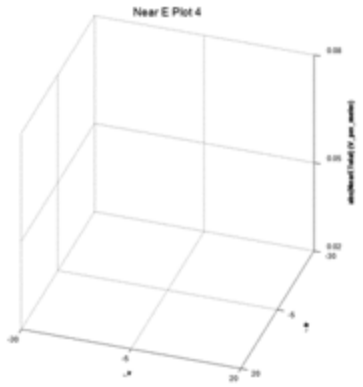
Selecting a bar in the plot shows the customizable attributes of bars in a 3D Rectangular Bar plot.

The customizable attributes of the bars in a 3D Rectangular Bar plot include:

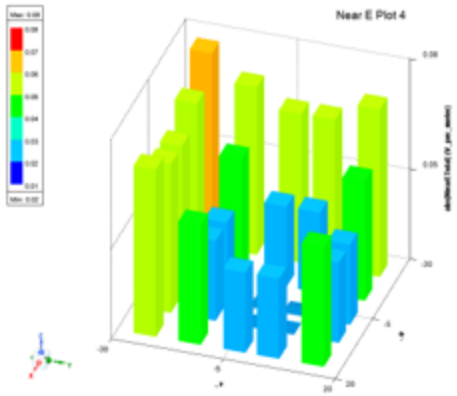
- Transparency



- Filled

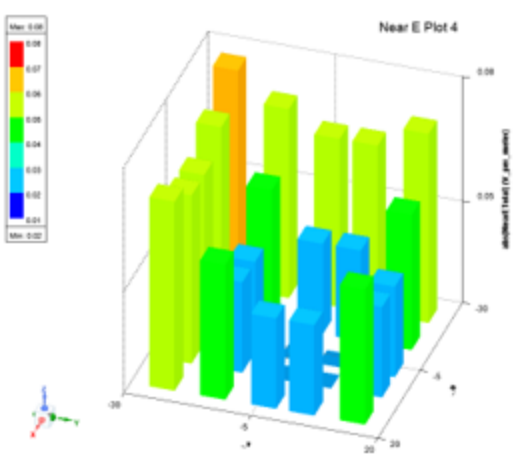


Filled [ ]

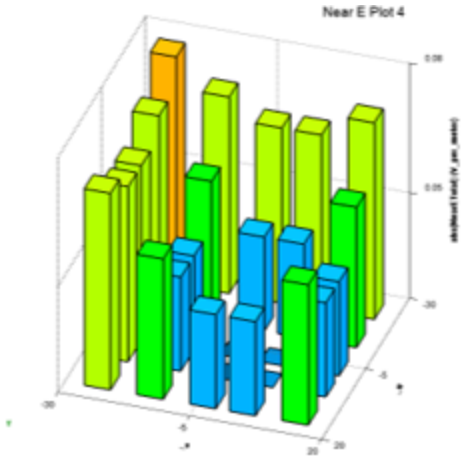


Filled [√]

- Show Outline



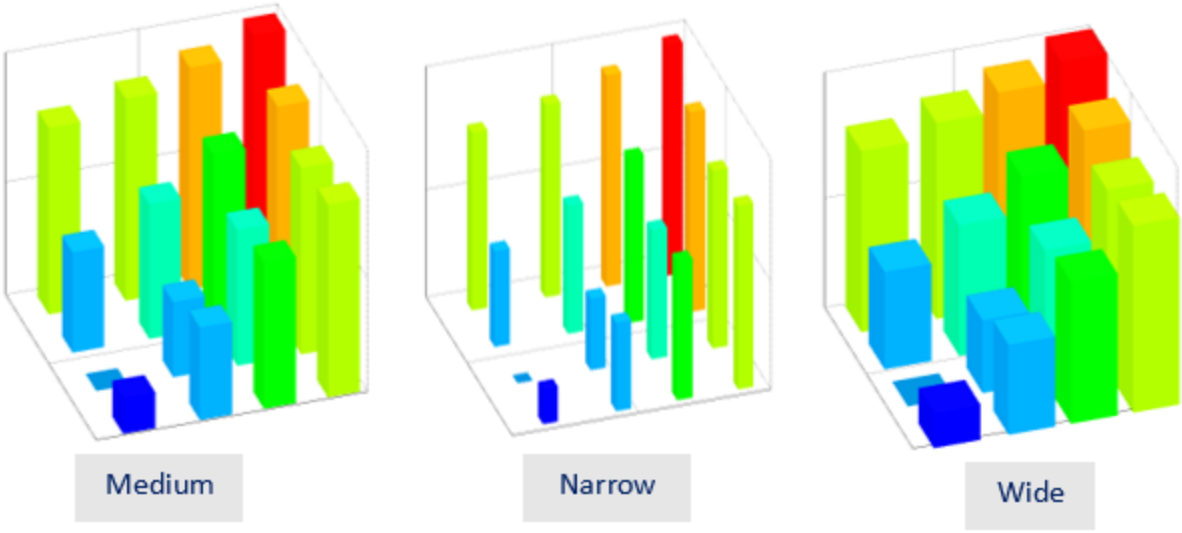
Show Outline []



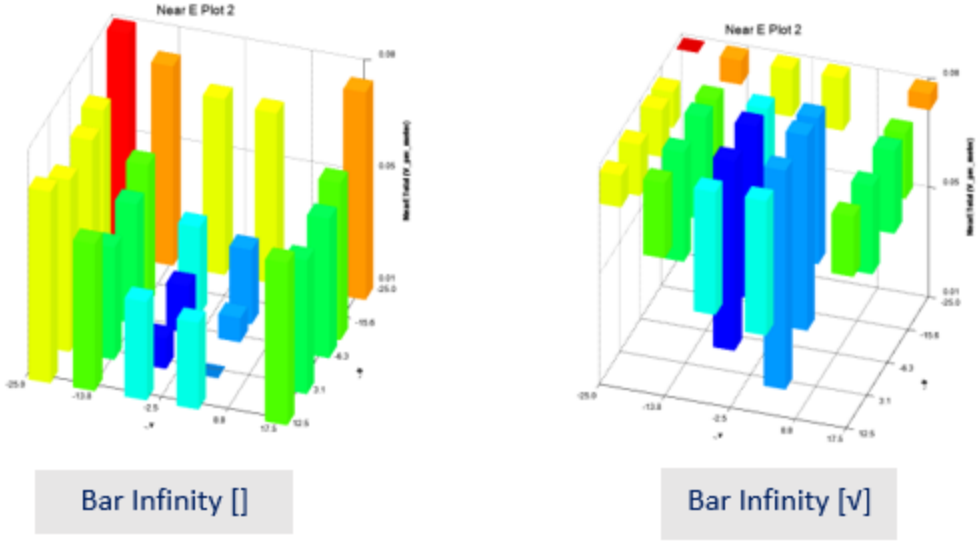
Show Outline [√]

- Outline Color (customizable, see plot above)
- Outline Width (customizable, see plot above)

- Bar Width, controls the thickness of the bar. Medium (default), Wide and Narrow are the width that equals to ½, ¾ and ¼ of the min spacing between adjacent points respectively.

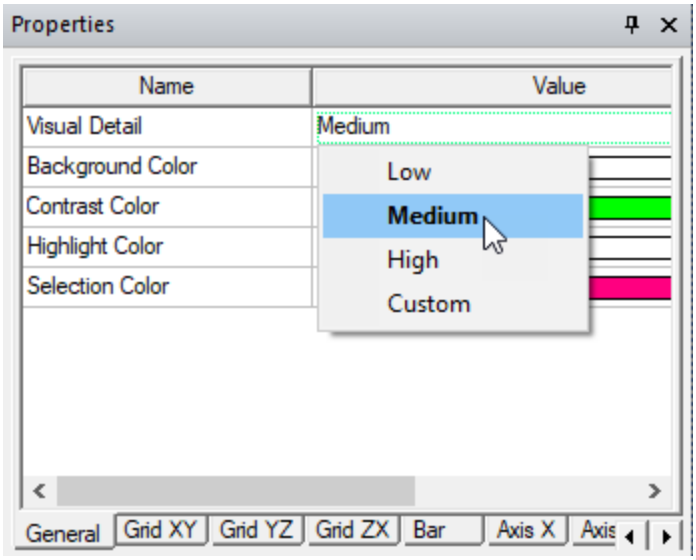


- Bar Infinity represents the third coordinate (z) using the bottom of the bar.



Controlling Visual Detail in a 3D Rectangular Bar Plot

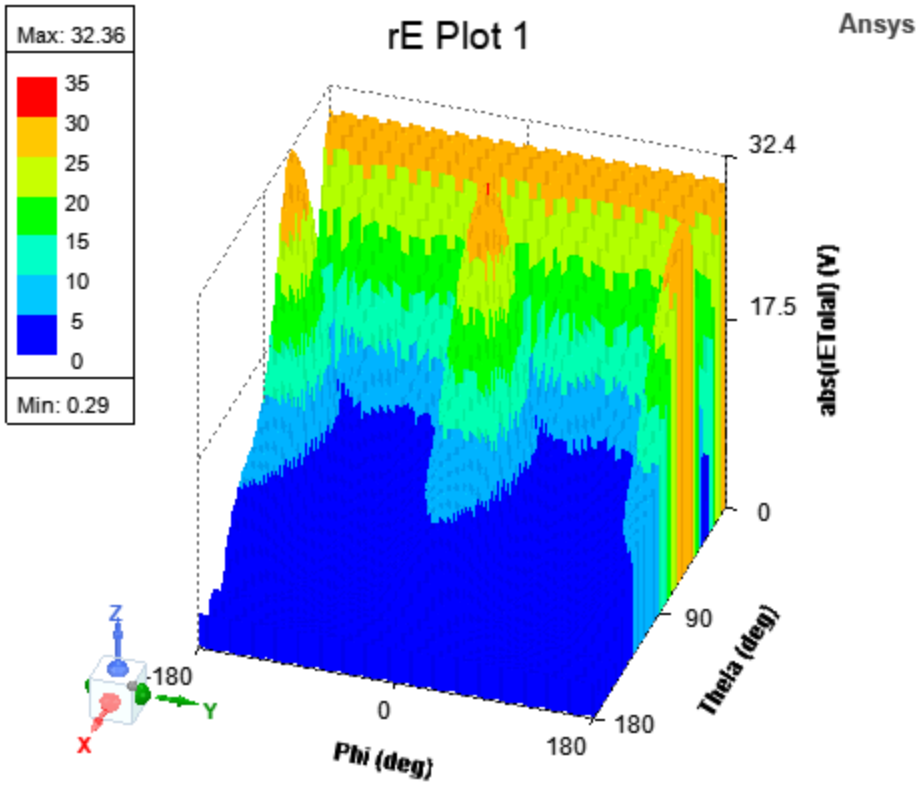
If a particular plot seems busy with information, you can edit plot properties, such as Axis and Grid Attributes for discrete levels of visual detail to improve readability. Double-click anywhere on a plot to display the Properties dialog box. The Visual Detail property on the General tab also provides control suited to different screen and plot sizes.



The Visual Detail menu has four options: Low, Medium (the default), High, and Custom. If you select any Visual Detail, the 3D plot is rendered according to the selected Visual Detail level and the properties reflect the values chosen for the selected visual detail level. From this predefined visual detail level, if you modify any properties, Visual Detail is automatically set to Custom (or to another predefined visual detail level if the edits happen to match the settings for that level).

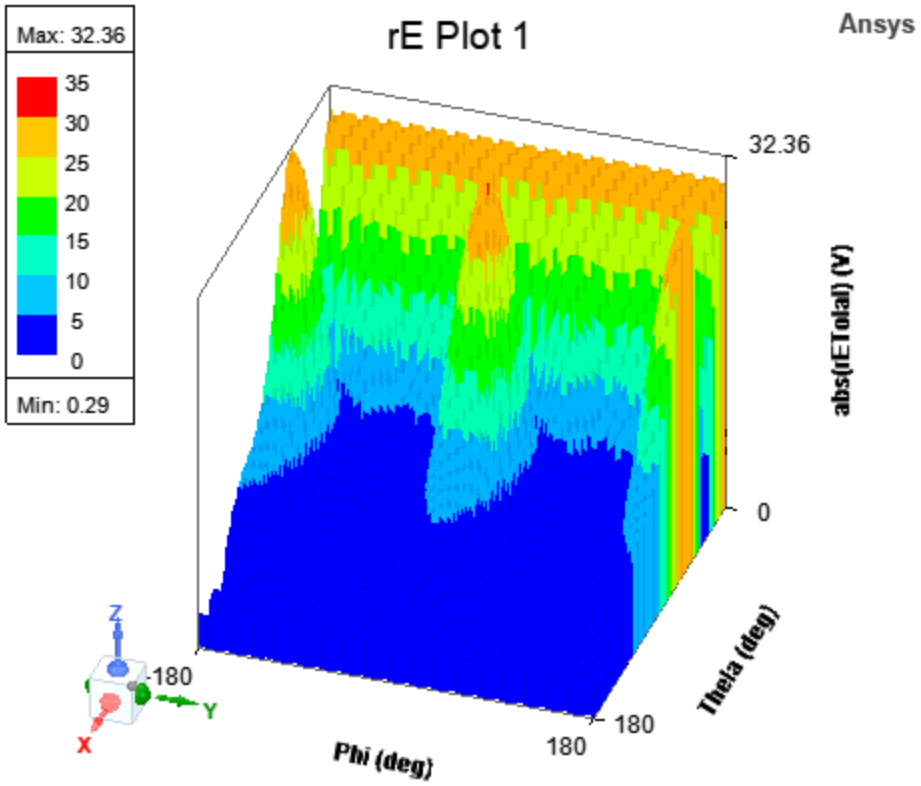
You can also manually set Visual Detail to Custom. In such a case, Custom will inherit property values corresponding to the previous level. This ensures that you can customize settings starting from a baseline provided by the preconfigured Low, Medium or High Visual Detail levels.

3D Rectangular Plot with Medium Visual Detail



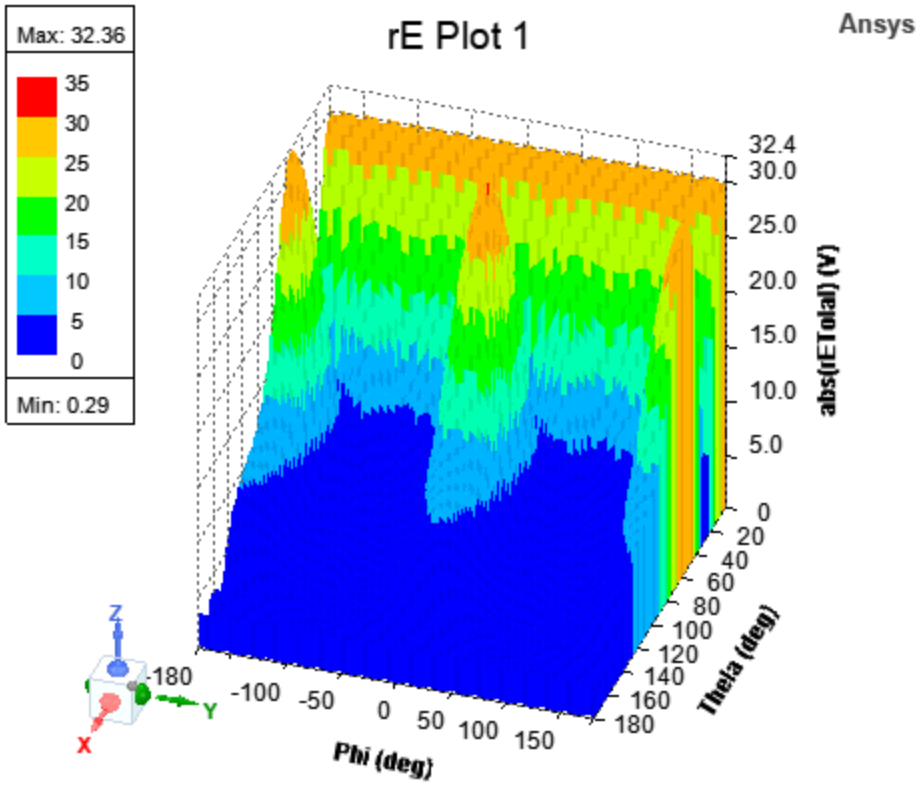
On creation, a 3D Rectangular Plot has Visual Detail set to Medium and looks and feels as shown above. Specifically, under Medium Visual Detail level, a 3D Rectangular Plot has 3 ticks per axis (X, Y, Z axis) which will show min, max and middle value. This setting also shows axes labels.

3D Rectangular Bar Plot with Low Visual Detail



With Visual Detail set to Low, a 3D Rectangular Plot shows axes with 2 ticks corresponding to min and max values. It also shows axes labels and grid borders.

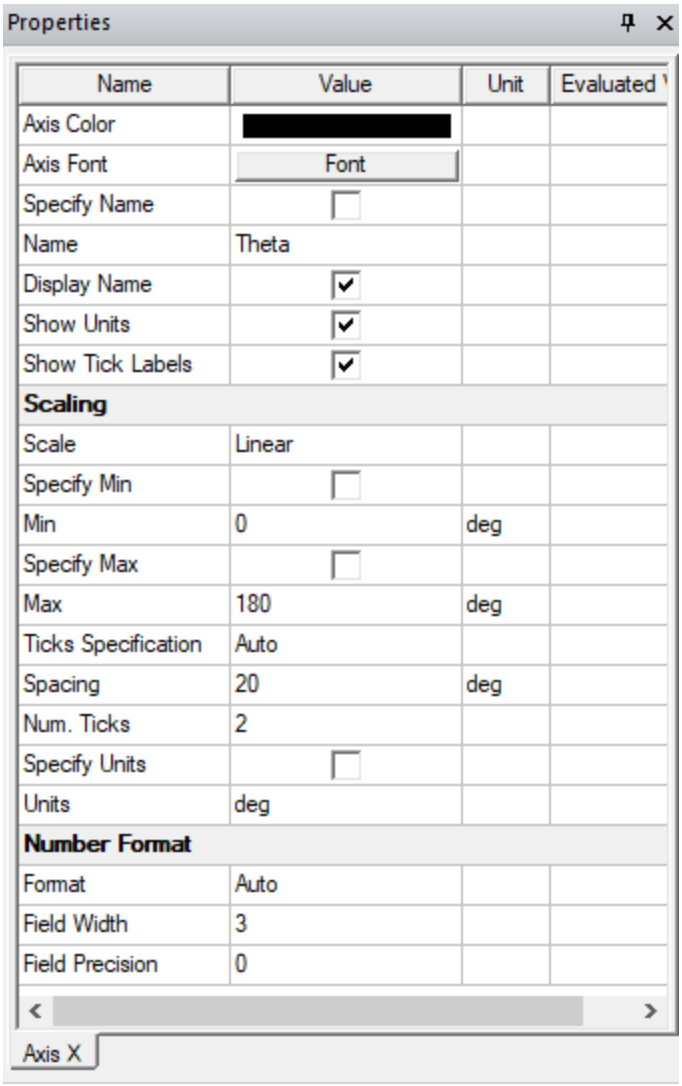
3D Rectangular Bar Plot with High Visual Detail



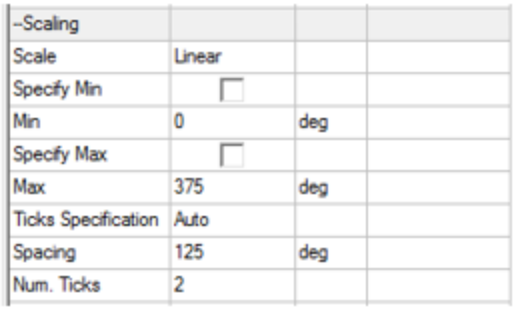
With Visual Detail set to High, a 3D Rectangular Plot shows all Cartesian axes and grids together with all ticks and axes labels.

Axis Properties: Ticks Specification and Num. Ticks

Ticks Specification is available on Axis properties, as shown below:



Ticks Specification is a menu with possible values as Auto, Spacing, and Num. Ticks, with Auto being the default value. If Ticks Specification is Auto, then a spacing value is automatically calculated and used to calculate and display the tick labels. **Spacing** shows the calculated value, and Num. Ticks shows the number of ticks based on this spacing value, as shown below:



You can edit the **Spacing** field when Ticks Specification is set to Spacing; otherwise, it is read only.

You can edit the **Num. Ticks** field when Ticks Specification is Num. Ticks; otherwise, it is read only.

Valid Num. Ticks are between 0 and 100, including 0 and 100. If you enter an invalid value, an error message is shown. If you enter a spacing value that results in number of ticks greater than 100, then an appropriate value is shown.

- If Num. Ticks is 0, then no ticks are shown on the axis.
- If Num. Ticks is 1, then only the max value tick is shown on the axis.
- If Num. Ticks is 2, then only the min and max value ticks are shown on the axis.
- If Num. Ticks is greater than 2, then evenly spaced ticks (including min and max) are shown on the axis.

> **Note:**
>
> With the addition of the Ticks Specification property to Axis properties, the **Specify Spacing** property was removed as an Axis property.
>
> - If an R18.0 or R18.1 project is opened with **Specify Spacing** as Unchecked, Ticks Specification is set to Auto.
> - If an R18.0 or R18.1 project is opened with **Specify Spacing** as Checked, Ticks Specification is set to Spacing.

Related Topics:

Sweeping a Variable

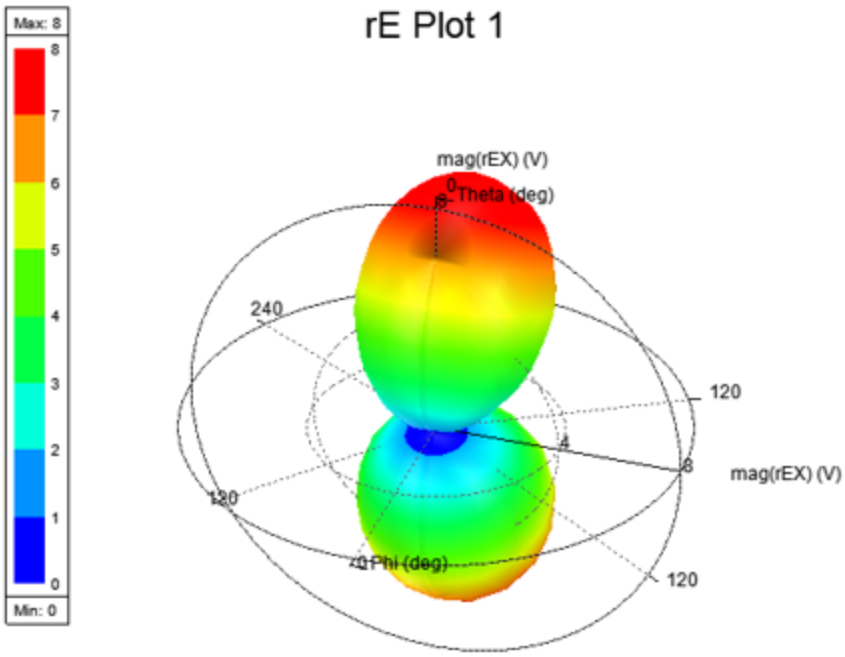
Working with Traces

Adding Characteristics to a Trace

Using the Orientation Gadget

Creating 3D Polar Plots in RMxprt

A 3D polar plot is a 3D circular chart divided by the spherical coordinates R, theta, and phi, where R is the radius, or distance from the origin, theta is the angle from the x-axis, and phi is the angle from the origin in the z direction.



Interacting with 3D Polar Plots

You can Rotate, Zoom and Pan a plot.

Clicking on a plot entity selects it, highlighting the selected entity in bold.

Double-clicking anywhere in the plot brings up the Properties dialog box. The properties are grouped appropriately under various tabs, which correspond to plot entities:

- General: For general plot properties such as Visual Detail level and background color
- Header: Properties related to plot Header/Title
- Axis Phi: Properties related to the circular axis which is in XY plane
- Axis Theta: Properties related to the circular axis which is in YZ plane
- Axis Rho: Properties related to the radial axis
- Grid phi-rho-theta(0): Properties related to phi-rho grid at theta = 0 (XY plane)
- Grid theta-rho-phi(90): Properties related to theta-rho grid at phi = 90 (YZ plane)
- Color Key: Properties related to the color key, including borders, background, Min and Max, as well as number format and precision.
- Contour: Properties related to contouring of all curves/surfaces
- Surface: Properties related to the curve

Selecting a property also displays its properties in the Property window. You can edit the properties to customize the appearance of the plot. See "Controlling Visual Detail in a 3D Plot" on page 26-178.

Creating a 3D Rectangular Plot

1. Click **RMxprt>Results>Create RMxprt Report>3D Rectangular Plot** .
2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   a. Solution field with a drop down selection list. This lists the available solutions, whether sweeps or adaptive passes.
   b. Domain field with a pull down selection list containing options for plotting vs ElectricDegree, Speed, or a user-selected Parameter.
3. Under the **Z** tab, specify the information to plot along the z-axis:
   a. In the **Category** list, click the type of information to plot.
   b. In the **Quantity** list, click the value to plot.
   c. In the **Function** list, click the mathematical function of the quantity to plot.
   d. The **Value** field displays the currently specified Quantity and Function. You can edit this field directly.

Note	Color shows valid expression.

   e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.
4. On the **Trace** tab, **Y** (**Secondary Sweep**) lines, specify the information to plot along the y-axis in one of the following ways:
   - Select the sweep variable to use from the Secondary Sweep drop down list.
   - If sweeps are available, you can also select the browse [...] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time sweeps), or **Use all values** (the default setting). The quantity will be plotted against the primary sweep variable listed.
5. On the **Trace** tab, **X** (**Primary Sweep**) lines, specify the information to plot along the x-axis in one of the following ways:
   - Select the sweep variable to use from the Primary Sweep drop down list.
   - If sweeps are available, you can also select the browse [...] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time sweeps), or **Use all values** (the default setting). The quantity will be plotted against the primary sweep variable listed.
6. On the **Families** tab, confirm or modify the sweep variables that will be plotted.
7. Click **New Report.**

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog box. The default name is based on the Report Category you selected, (for example, Force Plot *n* or Output Variables Plot *n*). You can edit the plot names in the project tree and the plot header text in the report synchronizes.

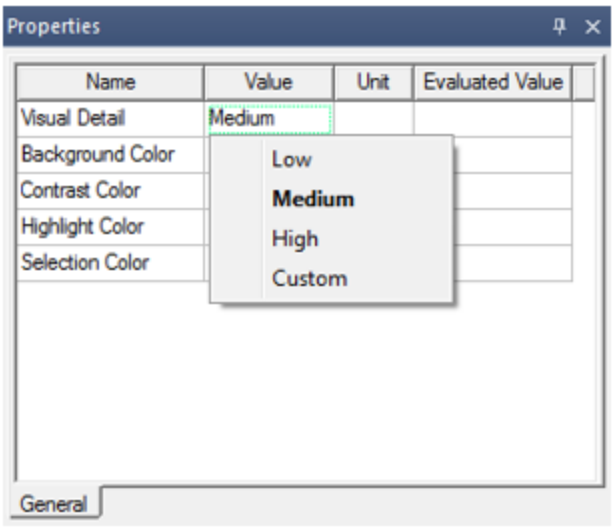
   The function of the selected quantity or quantities will be plotted against the values you specified on an x-y-z graph. The plot is listed under **Results** in the project tree.When you

select the traces or plots, axis or grid labels, plot header, color key, or variable labels, their properties are displayed in the Properties window. The properties for each plot element can be edited directly to modify the plot content and appearance. See Modifying the Background Properties of a Report.

8. Optionally, add another trace to the plot by following the procedure above, using **Add Trace rather than New Report**.

Controlling Visual Detail in a 3D Plot

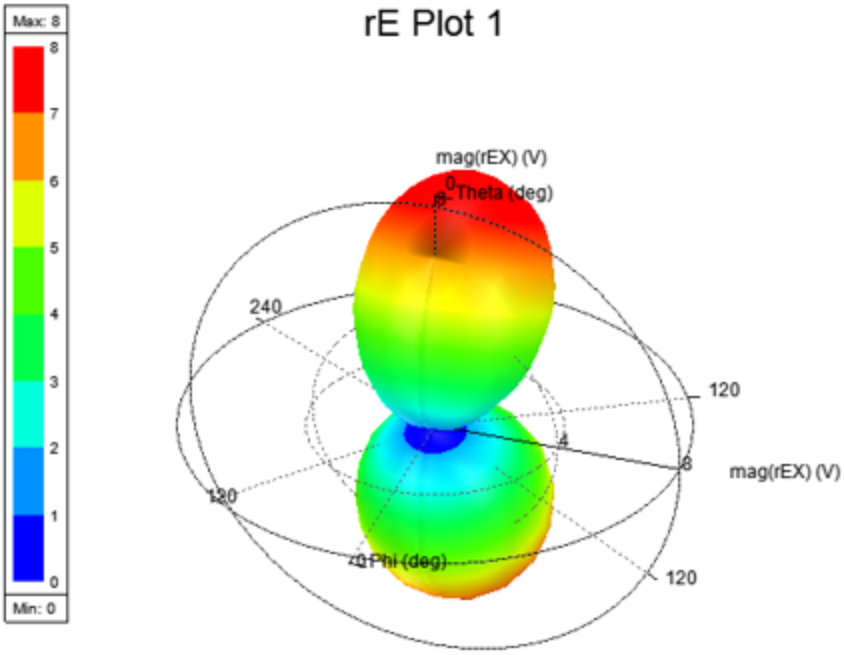
If a particular plot seems busy with information, you can edit plot properties, such as Axis and Grid Attributes for discrete levels of visual detail to improve readability. Double-click anywhere on a plot to display the Properties dialog box. The Visual Detail property on the General tab also provides control suited to different screen and plot sizes.



The Visual Detail menu has four options: Low, Medium (the default), High, and Custom. If you select any Visual Detail, the 3D plot is rendered according to the selected Visual Detail level and the properties reflect the values chosen for the selected visual detail level. From this predefined visual detail level, if you modify any properties, Visual Detail is automatically set to Custom (or to another predefined visual detail level if the edits happen to match the settings for that level).

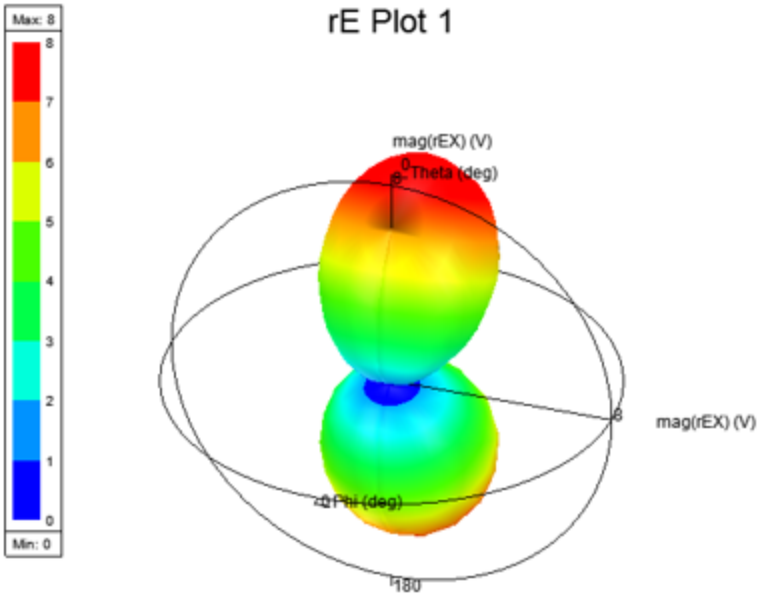
You can also manually set Visual Detail to Custom. In such a case, Custom will inherit property values corresponding to the previous level. This ensures that you can customize settings starting from a baseline provided by the preconfigured Low, Medium or High Visual Detail levels.

3D Polar Plot with Medium Visual Detail



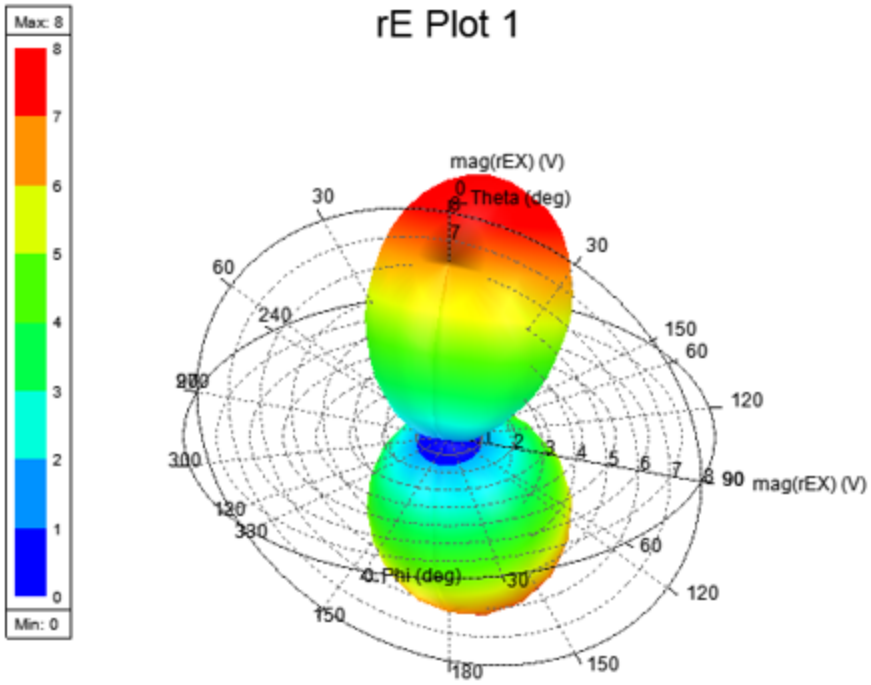
On creation, a 3D Polar Plot has Visual Detail set to Medium and looks and feels as shown above. Specifically, under the Medium Visual Detail level, 3D Polar Plot has 3 ticks per axis (phi, theta and rho axis) which show min, max and middle value. This setting also shows axes labels.

3D Polar Plot with Low Visual Detail



With Visual Detail set to Low, 3D Polar Plot does not show polar grids or grid lines. It only shows axes with 2 ticks corresponding to min and max values. This setting also renders axis labels.

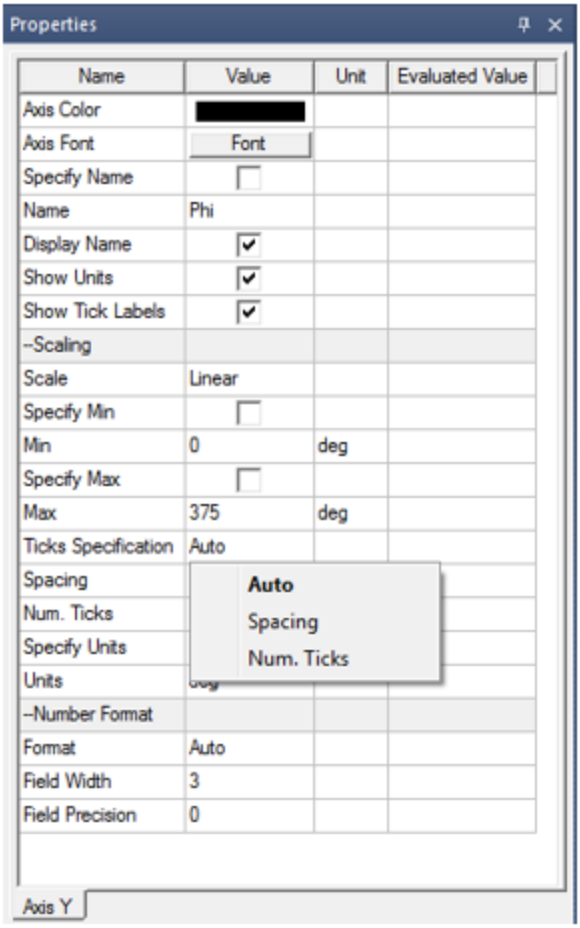
3D Polar Plot with High Visual Detail



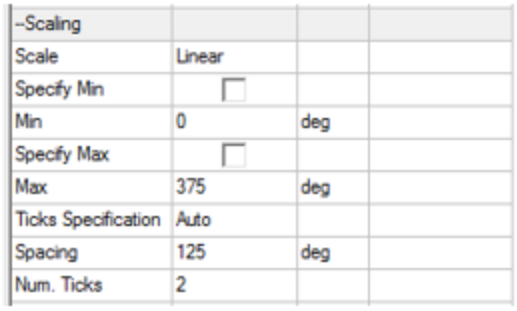
With Visual Detail set to High, a 3D Polar Plot shows all polar axes and grids together with all ticks and axes labels. This is ideal for large plot sizes.

Axis Properties: Ticks Specification and Num. Ticks

Ticks Specification is available on Axis properties, as shown below:



Ticks Specification is a menu with possible values as Auto, Spacing, and Num. Ticks, with Auto being the default value. If Ticks Specification is Auto, then a spacing value is automatically calculated and used to calculate and display the tick labels. **Spacing** shows the calculated value, and Num. Ticks shows the number of ticks based on this spacing value, as shown below:



You can edit the **Spacing** field when Ticks Specification is set to Spacing; otherwise, it is read only.

You can edit the **Num. Ticks** field when Ticks Specification is Num. Ticks; otherwise, it is read only.

Valid Num. Ticks are between 0 and 100, including 0 and 100. If you enter an invalid value, an error message is shown. If you enter a spacing value that results in number of ticks greater than 100, then an appropriate value is shown.

- If Num. Ticks is 0, then no ticks are shown on the axis.
- If Num. Ticks is 1, then only the max value tick is shown on the axis.
- If Num. Ticks is 2, then only the min and max value ticks are shown on the axis.
- If Num. Ticks is greater than 2, then evenly spaced ticks (including min and max) are shown on the axis.

Note	With the addition of the Ticks Specification property to Axis properties, the **Specify Spacing** property was removed as an Axis property. • If an R18.0 or R18.1 project is opened with **Specify Spacing** as Unchecked, Ticks Specification is set to Auto. • If an R18.0 or R18.1 project is opened with **Specify Spacing** as Checked, Ticks Specification is set to Spacing.

Related Topics

Sweeping a Variable

Working with Traces

Creating Data Tables in RMxprt

A data table is a grid with rows and columns that displays, in numeric form, selected quantities against a swept variable or other quantities.

1. Click **RMxprt>Results>Create RMxprt Report>Data Table**.

   The **Report** dialog box appears.
2. In the **Context** section make selections from the following field or fields, depending on the design and solution type.
   a. Solution field with a drop down selection list. This lists the available solutions, whether sweeps or adaptive passes.
   b. Domain field with a drop down selection list.
3. Under the **Trace** tab, **Y** component section, select the quantity you are interested in and its associated function:
4. Under the **Y** tab, select the quantity you are interested in and its associated function:
   a. On the **Category** drop down list, click the type of information to plot.
   b. On the **Quantity** list, click the values to plot. Use CTRL-click to make multiple selections.
   c. In the **Function** list, click the mathematical function to use for the quantity.

d. The **Value** field displays the currently specified Quantity and Function. You can edit this field directly.

Note	Color shows valid expression.

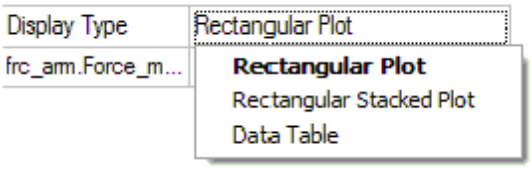
e. **Range Function** button -- opens the **Set Range Function** dialog box. This applies currently specified Quantity and Function.

5. On the **Trace** tab, **X (Primary Sweep) line,** select the sweep variable from the drop down list. By default **All** of the chosen sweep's values are used. You can also select the browse [...] button to display a dialog box that lets you select particular sweep values, specify a range of sweep values (for Time sweeps), or **Use all values** (the default setting).
6. On the **Families** tab, confirm or modify the sweep variables that will be plotted.
7. Click **New Report**.

   This creates a new report in Project tree, displays the report with the defined trace, and enables the **Add Trace** button on the **Report** dialog box. The default name is based on the Report Category you selected, for example, Force Plot *n* or Output Variables Plot *n*. You can edit the plot names in the project tree and the plot header text in the report synchronizes.

   The Y quantity will be listed at each variable value or additional quantity value you specified. The data table is listed under **Results** in the project tree. The plot is listed under **Results** in the project tree and the traces are listed under the plot. When you select the traces or plots, their properties are displayed in the Properties window. These properties can be edited directly to modify the plot.
8. Optionally, add another trace to the plot by following the procedure above, using **Add Trace** rather than **New Report**.

You can also modify the display type of an existing plot from the Properties dialog for that plot. Select the Report icon in the Project tree to display the Properties dialog box. Selecting the Display Type field displays a menu with selections available for that plot.



Once you make a selection, the plot display updates for the current selection.

If you choose to print a data table:

- Selecting print "All" prints the whole table for current data page (if there is more than one data page).
- Selecting print "Pages" prints user-specified pages.
- If the table is bigger than the screen view (that is, it has a scroll bar), printing first scrolls right, prints until no more scrolling occurs, and then scrolls down.
- The Page number appears at the bottom of the page, aligned at center.

- The table layout of each page follows the screen, but without the scroll bar being printed, and no data page bar as on screen.

Related Topics

Sweeping a Variable
Working with Traces

Working with Traces in RMxprt

A trace in a 2D or 3D report defines one or more curves on a graph. A trace in a data table defines part of the displayed matrix of text values.

The values used for a plot's axes can be variables in the design or functions and expressions based on the design's solutions. If you have solved one or more variables at several values, you can "sweep" over some or all of those values, resulting in a curve in 2D or 3D space.

A report can include any number of traces and, for rectangular graphs, up to four independent y-axes.

In general, to add a trace to a report:

1. In the **Traces** dialog box, specify the information you want to plot along the appropriate axes.
2. Click **Add Trace**.

   A trace is added to the traces list at the top of the **Traces** dialog box. The trace represents the function of the quantity you selected and will be plotted against other quantities or swept variable values. Each column lists an axis on the report and the information that will be plotted on that axis.

   You can modify the information to be plotted by typing the name of the quantity or sweep variable to plot along an axis directly in the boxes.

   The trace will be visible in the report when you click **Done**.

Note	• The **Traces** dialog box can be accessed via the **Create Report**dialog box**.** • You can also Add Trace Characteristics to 2D reports.

Removing Traces in RMxprt

You can traces from the traces list in the following ways:

To *remove one trace* from the report:

- Select the trace you want to remove from the traces list, and then click **Remove Trace**.

To *remove all traces* from the report:

- Click **Remove All Traces**.

Related Topics

Working with Traces

Replacing Traces in RMxprt

To replace a trace in the traces list with a different trace definition:

1. Select the trace you want to remove from the traces list.
2. In the **Traces** dialog box, specify the information you want to plot along the appropriate axes.
3. Click **Replace Trace**.

   The trace you selected is removed, and the new trace information you specified replaces it in the traces list.
4. Click **Done**.

Related Topics

Working with Traces

Adding Blank Traces in RMxprt

To add a blank trace to the traces list:

- Click **Add Blank Trace**.

You can now type the quantities to plot in the appropriate axes boxes.

Related Topics

Working with Traces

Sweeping a Variable in a Report in RMxprt

In RMxprt, a swept variable is an intrinsic, project, or design variable that typically has more than one value. From the **Traces** dialog box, you can plot any calculated or derived quantity against one or more of the swept variable's values.

When you click the Sweeps tab in the **Traces** dialog box, the first sweep variable listed is the "primary sweep". If you are creating a 3D report, the second sweep variable listed is the "secondary sweep". Any additional sweep variables are represented as additional curves on the graph.

To modify which variable is the primary sweep variable:

- Click the **Name** box for the primary sweep variable, and then click the variable name you want to be the primary sweep variable.

To modify the secondary sweep variable or any additional sweep variable, follow the same procedure.

To modify the values that will be plotted for a variable:

1. Click a variable in the table.

   To the right, all of the possible values for the selected variable are listed.
2. Select **All Values**.

   All of the selected variable's values are plotted.
   - Alternatively, clear **All Values** and select the specific values to plot against the selected quantity.

Selecting a Function in RMxprt

The value of a quantity being plotted depends upon its mathematical function, which you select from the **Function** list in the **Report** dialog box. The available, valid functions depend on the type of quantity (real or complex) that is being plotted. The function is applied to the quantity which is implicitly defined by all the swept and current variables. For example, "Volume(Stator)" is the volume of the model object named Stator for every swept combination of variables ("height" and so forth).

These functions can also be applied to previously specified Quantities and Functions as **Range Functions** when using the Set Range Function dialog.

Some of these functions can operate along an entire curve. These are: min, max, integ, rms, pk2pk, cang_deg and cang_rad.

You can select from the following functions in the **Trace** tab **Function** list, or type them directly into the Y or X field, if necessary:

abs	Absolute value of the simulation quantity which results in a number that is always positive.
acos	Arc cosine i.e. the inverse function of a cosine.
acosh	Inverse hyperbolic arc cosine.
ang	Magnitude of an angle.
ang_deg	Angle (phase) of a complex number, cut at +/-180.
ang_rad	Angle in radians.
arg	Argument of a complex number. It is the angle the complex number makes with the positive x axis. Same as **ang_deg**.
asin	Arc sine i.e. inverse function of sine.
asinh	Inverse hyperbolic sine.
atan	Arc tangent. Takes a tangent value as an argument. Because there are two angles in a circle that can have the same tangent value, and atan can return only one value, it returns a value in the range between -90 degrees and +90 degrees (or between -pi/2 and pi/2 in radians).

atanh	Inverse hyperbolic tan.
atan2	A two-argument version of the **atan** function. Takes the y and x coordinates (including sign information) of a point as arguments and returns the angle from the X-axis. Can return angle values for the full circle (-180 degrees to +180 degrees or -pi to +pi in radians).
avg	Returns the average of the values of the selected quantity. avg = (Area between the curve and the X-axis) / (X length of the curve)
avgabs	Returns the mean of the absolute value of the selected quantity.
bandwidth	Returns the 3dB bandwidth of the selected simulation quantity. For bandwidth, the calculation is based on 3dB below the maximum peak.
cang_deg	Cumulative angle (phase) of the first parameter (a complex number) in degrees, along the second parameter (typically sweep variable). Returns a double precision value cut at +/-180.
cang_rad	Cumulative angle of the first parameter in radians along a second parameter (typically a sweep variable). Returns a double precision value.
cmplx(*re*, *im*)	A complex number, where *re* is the real part and *im* is the imaginary part.
conjg	Conjugate of the complex number.
cos	Cosine.
cosh	Hyperbolic cosine.
crestfactor	Returns the crest factor (peak/RMS) for the selected quantity.
cum_integ	The cumulative integral function returns a set of values that have the same length as the original set of points (the first element will always be zero). Element *I* of the set returned by cum_integ is the integral of elements 1 through *I* of the original data set.
cum_sum	The cumulative sum function returns a data set that has the same length as the original set of points. Element *I* of the set returned by cum_sum is the sum of elements 1 through *I* of the original data set.
dB(x)	20*log10(|x|) to base 10.
dBc	Decibels relative to the carrier. It is the power ratio of the signal to a carrier signal. Gives the relative signal strength.
dBm(x)	10*log10(|x|) +30.
dBW(x)	10*log10(|x|).
dB10	10*log(|x|) to base 10.
dB10normalize	10*log [normalize(mag(x))].
dB20	20*log(x) to base 10.
dB20normalize	20*log [normalize(mag(x))].
deadtime	Obtains the latest time when the qtyl is within a tolerance of zero.

delaytime	Obtains the time from zero to 50% of the target point.
degel	Conversion from degrees electrical to seconds with respect to Hz.
deriv	Derivative of a given parameter.
distortion	Returns the total distortion for the selected simulation quantity and an additional argument frequency, which is the frequency in Hz at which to calculate the fundamental RMS of the simulation quantity.
even	Returns 1 if integer part of the number is even; returns 0 otherwise.
exp	Exponential function (the natural anti-logarithm) of the simulation quantity.
formfactor	Returns the form factor (RMS/Mean Absolute Value) for the selected quantity.
fundamentalmag	Returns the RMS value of the fundamental frequency for the selected quantity, and an additional argument, Frequency, which specifies the fundamental frequency.
gaincrossover	Returns the gain crossover frequency (where the gain is 0 dB) of the selected simulation quantity in Hz.
gainmargin	Returns the gain margin in dB at the phase crossover frequency of the selected simulation quantity. It also requires a reference simulation quantity to which the measured quantity is compared and the AC magnitude and phase angle of the reference quantity. These are entered as the arguments Reference Channel, Base Source Magnitude, and Base Source Angle.
iae	Returns the integral of the absolute deviation of the selected quantity from a target value that is entered via the additional argument. To use this function, you need to open the Add Trace Characteristics dialog and select the Error category.
if	if(cond_exp,true_exp, false_exp).
im	Imaginary part of the complex number.
int	Truncated integer function.
integ	Integral of the selected quantity. Uses trapezoidal area.
integabs	Absolute value of integral.
ise	Returns the integral of the squared deviation of the selected quantity from a target value that is entered via an additional argument. To use this function, you need to open the Add Trace Characteristics dialog and select the Error category.
itae	Returns the time-weighted squared deviation of the selected quantity from a target value that is entered via an additional argument. To use this function, you need to open the Add Trace Characteristics dialog and select the Error category.
itse	Returns the time-weighted squared deviation of the selected quantity from a target value that is entered via an additional argument. To use this function, you need to open the Add Trace Characteristics dialog and

	select the Error category.
j0	Bessel function of the first kind (0^{th} order).
j1	Bessel function of the first kind (1^{st} order).
jn	Bessel function of the first kind (nth order).
ln	Natural logarithm.
log	Natural logarithm (same as ln).
log10	Logarithm base 10.
lowercutoff	Returns the lower 3dB frequency of the selected simulation channel in Hertz.
lsidelobeX	The 'x' value for the left side lobe: the next highest value to the left of the max value.
lsidelobeY	The 'y' value for the left side lobe: the next highest value to the left of the max value.
mag	Magnitude of the complex number.
max	Returns maximum value of the simulation quantity.
max_swp	Returns maximum value of a sweep.
max2	Maximum value of the two simulation quantities. For example, **max2(a,b)** will plot maximum of **a** and **b** for a particular instance.
mean	Returns the average in the set of quantities selected. mean = sum(all y-value) / (number of y-values)
min	Returns the minimum value of the simulation quantity.
min_swp	Returns the minimum value of a sweep.
min2	Minimum value of the two simulation quantities. For example, **min2(a,b)** will plot minimum of **a** and **b** for a particular instance.
mod	Returns the modulus or absolute value of the simulation quantity.
nint	Nearest integer.
none	Returns null value.
normalize	Divides each value within a trace by the maximum value of the trace. ex. normalize(mag(x)).
odd	Returns 1 if integer part of the number is odd; returns 0 otherwise.
overshoot	Calculates peak overshoot given a threshold value and number of evenly spaced points over entire time range.
peakgain	Returns the peak value of gain of the selected simulation quantity in dB.
peakgainfreq	Returns the frequency in Hz at which the peak gain of the selected simulation quantity occurs.
polar	Coverts the complex number in rectangular co-ordinates to polar co-ordinates.
per	Returns the period of a simulation quantity.

phasecrossover	Returns the phase crossover frequency, at which the phase is -180 degrees, in Hz for the selected simulation quantity.
phasemargin	Returns the phase angle in degrees at the gain crossover frequency of the selected simulation quantity.
pk2pk	Peak to peak. Difference between max and min of the first parameter over the second parameter. Returns the peak-to-peak value for the selected simulation quantity.
pkavg	Returns the ratio of the peak to peak-to-average for the selected quantity.
pmax	Maximum period of the selected simulation quantity.
pmin	Minimum period of the selected simulation quantity.
pow	Raises x to the power of y; pow(x,y).
prms	Period Root Mean Square.
pulsefall9010	Returns the pulse fall time of the selected quantity according to the 90%-10% estimate.
pulsefront1090	Returns the pulse front time of the selected quantity according to the 10%-90% estimate.
pulsefront3090	Returns the pulse front time of the selected quantity according to the 30%-90% estimate.
pulsemax	Returns the pulse maximum from the front and tail estimates for the selected quantity.
pulsemaxtime	Returns the time at which the maximum pulse value of the selected quantity is reached.
pulsemin	Returns the pulse minimum from the front and tail estimates for the selected quantity.
pulsemintime	Returns the time at which the minimum pulse value of the selected quantity is reached.
pulsetail50	Returns the pulse tail time of the selected quantity from the virtual peak to 50%.
pulsewidth5050	Returns the pulse width of the selected quantity as measured from the 50% points on the pulse front and pulse tail.
pwl	Piecewise Linear.
pwl_periodic	Piecewise Linear for periodic extrapolation on x.
pwlx	Piecewise Linear x with linear extrapolation on x.
pw_minus	Pulse width of the first negative pulse.
pw_minus_avg	Returns the average of the negative pulse width input stream.
pw_minus_max	Returns the maximum pulse width of the negative pulse of input stream.
pw_minus_min	Returns the minimum pulse width of the negative pulse of input stream.
pw_minus_rms	RMS of the negative pulse width input stream.
pw_plus	Pulse width of the first positive pulse.

pw_plus_avg	Average of the positive pulse width input stream.
pw_plus_max	Max. Pulse width of the positive pulse of input stream.
pw_plus_min	Min. Pulse width of the positive pulse of input stream.
pw_plus_rms	RMS of the positive pulse width input stream.
re	Real part of the complex number.
rect	Converts the complex number in polar to rectangular co-ordinates.
rem	Fractional part of the selected simulation quantity i.e. remainder.
ripple	Returns the ripple factor (AC RMS/Mean) for the selected quantity.
risetime	Obtains the time taken to go from 10% to 90% of target point.
rms	Returns the root mean square value of the selected quantity.
rmsAC	Returns the AC RMS for the selected quantity.
root	nth root function.
rSidelobeX	Returns the X value of right side-lobe occurrence.
rSidelobeY	Returns the Y value of right side-lobe occurrence.
settlingtime	Returns the latest time at which the value of the selected simulation quantity fell outside its tolerance band. The target value of the quantity and the +/- bandwidth of the tolerance band are the additional args.
sgn	Sign extraction.
sin	Sine.
sinh	Hyperbolic sine.
slidingmean	Returns the moving average value of the selected simulation quantity (specified by the first argument). The average is calculated over a period (specified by the second argument).
slidingrms	Returns the moving RMS value of the selected simulation quantity (specified by the first argument). The RMS value is calculated over a period (specified by the second argument).
sqr	Square of the selected simulation quantity.
sqrt	Square root of the selected simulation quantity.
stddev	Returns the standard deviation of given values.
sum	Returns the sum of the given values.
tan	Tangent.
tanh	Hyperbolic tangent.
undershoot	Calculates peak undershoot given a threshold value and number of evenly spaced points over entire time range.
uppercutoff	Returns the upper 3dB frequency of the selected simulation channel in Hz.
variance	Calculates the variance of the given values.
XAtYMax	Threshold crossing time: report first time (x value) at which an output

	quantity crosses YMax.
XAtYMin	Threshold crossing time: report first time (x value) at which an output quantity crosses a user definable threshold.
XAtYVal	Returns the X value at the first occurrence of Y value.
XWidthAtYVal	Returns the X width between the first 2 occurrence of Y value.
xdb10beamdwidth	Width between left and right occurrences of values 'x' db10 from max. Takes 'x' as argument (3.0 default). To use this function, you need to open the Add Trace Characteristics dialog and select the Radiation category.
xdb20beamwidth	Width between left and right occurrences of values 'x' db20 from max. Takes 'x' as argument (3.0 default). To use this function, you need to open the Add Trace Characteristics dialog and select the Radiation category.
YAtXMax	Returns the X value at maximum value of Y.
YAtXMin	Returns the Y value at minimum value of X.
YAtXVal	Returns the Y value at the first occurrence of X value.
y0	Bessel function of the second kind (0^{th} order).
y1	Bessel function of the second kind (1^{st} order).
yn	Bessel function of the second kind (nth order).

Selecting a Parameter, Variable, or Quantity to Plot in RMxprt

Each trace in a report includes a quantity that is plotted along an axis. The quantity being plotted can be a value that was calculated by RMxprt, such as L_{11}, a value from a calculated expression.

To select a parameter, variable, or quantity to plot:

1. In the **Traces** dialog box, select one of the following categories:

Variables	User-defined project or design variables.
Output Variables	Derived quantities RMxprt project or design variables, parameters or solution curves.

2. Select a quantity to plot from the **Quantity** list. The available quantities depend on the selected category and the setup of the design.

Creating Quick Reports in RMxprt

Following is the procedure for creating a quick report.

1. On the Project tree, select a **setup** or **sweep icon** of interest.
2. Right-click to display the shortcut menu and select **Quick Report**.
   The **Quick Report** dialog appears.
3. Select the one or more categories for the report from the list and click OK.

A rectangular plot for each selected category displays. The new plot or plots appear in the Project tree under the Results icon. The default Report Name that appears is derived from the report category specified in the **Quick Report** dialog box. You can edit the plot names in the project tree and the plot header text in the report synchronizes.

Related Topics

Creating Reports
Modifying Reports
RMxprt Quick Report Categories

RMxprt Quick Report Categories

When using the Quick Reports function for Solutions, the following report categories may be available depending upon the solution parameters requested, solution type, etc:

Category	Description
Coil Voltage	Report voltages in the machine coil.
Current	Report currents for each line or phase of the machine, source current, line current, armature current.
Flux Density	Report flux density in the machine air gap, flux linkages.
Induced Voltage	Report Induced Line and Phase voltages.
Misc.	Report miscellaneous quantities specific to the machine type such as power factor, torque to current ratio.
Percentage	Report machine efficiency.
Power	Report air gap power, output power.
Torque	Report cogging torque, output torque, magnet generated torque, induction torque.
Voltage	Report Line and Phase voltage.
Angle	Reports power factor angle.
Angular Speed	Reports angular speed.
Inductance	Reports air gap permeance.

Specifying RMxprt Winding Data

To define the winding data for an RMxprt machine

1. In the project tree, under **Machine**, open the folder that requires a winding, and double-click **Winding** to open the winding **Properties** dialog box.
   - For some machine types this would be **Machine>Rotor>Winding**, for others, **Machine>Stator>Winding**.
   - You can also enter values in the **Properties** section of the desktop without opening a separate window.
2. Specify the desired settings.
3. Click **OK** to close the **Properties** dialog box.

The specific properties available depend on the specific machine.

The following machine types have winding data available:

- Three-Phase Induction Motors (stator winding) and (rotor winding)
- Three-Phase Synchronous Machines (stator winding) and (rotor winding)
- Brushless PMDC Motors (stator winding)
- Adjust-Speed Synchronous Machines (stator winding)
- PMDC Motors (rotor winding)
- Switched Reluctance Motors (stator winding)
- Line-Start Permanent-Magnet Synchronous Motors (stator winding)
- Universal Motors (stator winding) and (rotor winding)
- General DC Machines (rotor winding)
- Claw-Pole Alternators (stator winding)
- Generic Rotating Machines

Setting the Winding Type

RMxprt can automatically arrange almost all commonly used single- or double-layer poly-phase ac windings provided all coils have the same number of turns. Users do not need to define coils one by one. For a double-layer winding, **RMxprt** can also handle the coils with half turns which are arranged in the order of even, odd, even, odd, …, as long as it is physically possible.

RMxprt also provides a very flexible tool **Winding Editor** in order for the users to design a variety of special winding types according to their own needs, such as compound single- and double-layer winding, big- and small-phase-spread variable-pole multiple-speed winding, sine-wave three-phase winding, and so forth. The **Winding Editor** is available to the following types of electric machines:

1. Three-phase induction motors
2. Single-phase induction motors
3. Three-phase synchronous motors and generators
4. Line-start permanent-magnet synchronous motors
5. Claw-pole alternators

6. Adjustable-speed permanent-magnet synchronous motors and generators
7. Brushless permanent-magnet DC motors

When you edit the AC winding of a new design for the first time, RMxprt creates a default winding arrangement based on the basic winding specifications: **Number of Phases**, **Number of Poles**, **Number of Slots**, **Winding Layers**, **Conductors per Slot**, and **Coil Pitch**. Then you can edit the winding configuration based on the default arrangement.

Winding Types Available for Machines

Use the **Winding Type** dialog to set the Winding type.

1. To display the **Winding Type** dialog double-click on the Winding property button.

   Passing the cursor over the buttons for the Winding types changes the graphic to show the available windings for the motor in the design. Choices differ depending on the motor. A Winding Editor selection does not have a graphic.

   Selections for the Three Phase Induction Motor (stator winding) and (rotor winding), Three-Phase Synchronous Machine (stator winding) and (rotor winding), Brushless Permanent Magnet DC Motor (stator winding), Adjust Speed Synchronous Machine (stator winding), Line Start PM Synchronous Motor (stator winding), and Claw Pole alternator (stator winding) include:

   - Editor - enable the Winding Editor
   - Whole Coiled
   - Half Coiled

   Selections for the DC Permanent Magnet Motor (rotor winding) and Universal Motor include (stator winding) and (rotor winding):

   - Lap
   - Wave

   Selection for the General DC Machines (rotor winding) include:

   - Lap
   - Wave
   - Frogleg

   Selections for single-phase induction motor include:

   - Editor - enable the Winding Editor
   - Lap - 90 deg phase belt 2-layer coil for both single and double layer
   - Sin_1 - first class sinusoidal coil for double layer only
   - Sin_2 - second class sinusoidal coil for double layer only

   The Switched Reluctance motor does not involve winding selections.

2. Select the **Winding Type** and click OK.

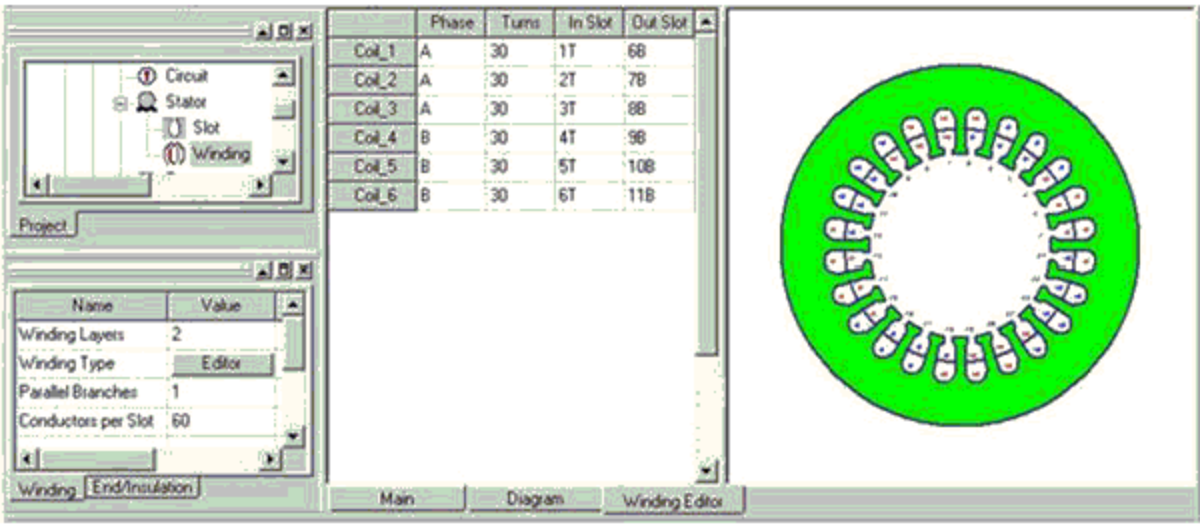
   This closes the window and sets the Winding Type property. If you select the Editor type, It also enables the **Machine>Edit Layout** command on the menu bar.

Enable the Winding Editor

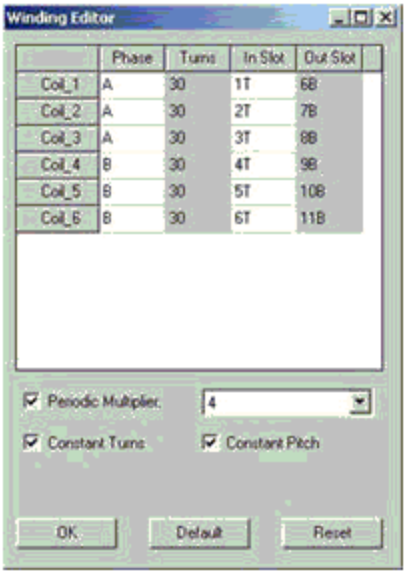
Setting the **Winding Type** property to **Editor** enables the command **Machine>Edit Layout** on the menu bar. To display the dialog box Winding Editor:

1. Open the Winding Properties window and set the Winding Type value to Editor. To do this, double-click on the button Winding Type value to display the Winding Type selection window.

2. Select Editor as the Winding Type and click OK. This closes the Winding Type selection window and sets the Winding Type Value to Editor. It also enables the command Machine>Winding>Edit Layout on the menu bar. Now the Machine Editor window displays the default winding arrangement.



3. Click **Machine>Winding>Edit Layout**. This displays the dialog box Winding Editor as shown. The dialog box Winding Editor includes functions that do not appear in the tab sheet Winding Editor in the RMxprt Machine Editor window.



4.

Edit Winding Configuration

Each row of the winding data table in the Winding Editor dialog box is identified with the coil index in the column Coil. This information is displayed in the tab sheet Winding Editor in the RMxprt Machine Editor window as well, but it is editable in the dialog box Winding Editor.

The winding data table contains four columns:

Phase	is for the phase to which the coil belongs.
Turns	is for the number of turns of the coil.
In Slots	is for the slot number with the coil side current flowing in ('flow-in-side' for short). If 2 Layers are specified in the Winding Properties window, the slot number ends with a "T" to show the top layer.
Out Slots	is for the slot number with the coil side current flowing out ("flow-out-side" for short). If 2 Layers are specified in the Winding Properties window, the slot number ends with a "B" to show the bottom layer.

Setting the Number of Winding Layers

To set the number of winding layers:

1. Open the **Winding Properties** window by double-clicking on the Winding icon in the properties window.
2. Use the drop-down menu in the Winding Layers field to set the number as 1 or 2.

   This sets the winding layers used in the winding.

   The number of layers selected makes a difference in the display of data in the **Winding Editor**.

Connecting and Disconnecting Windings

When you have specified the winding data, you can execute the following commands to automatically connect or disconnect the windings.

- **Machine>Winding>Connect All Coils**

  Upon executing, the graphical display in the main window shows the connections.
- **Machine>Winding>Disconnect All Coils**

  Upon executing, the graphical display in the main window updates to remove the connection.

Related Topics

View Winding Connections

Poly-phase Winding Editor

RMxprt provides a **Winding Editor** in order for users to design variety of special winding types according to their own needs, such as compound single- and double-layer winding, big- and small-phase-spread variable-pole multiple-speed winding, sine-wave three-phase winding, and so forth. The **Winding Editor** is available to the following types of electric machines:

- Three-phase Induction Motor
- Three-phase Synchronous Motor
- Three-phase Synchronous Generator
- Permanent-magnet Synchronous Generator
- Line-start Permanent-magnet Synchronous Motor
- Adjustable-speed Permanent-magnet Synchronous Motor
- Brushless Permanent-magnet DC Motor
- Claw-pole Alternator

You input data for **Number of Poles** in **the Machine Properties** window and data for the **Number of Slots** and Slot Type in the **StatorProperties** window. You set the **Number of Slots** in the **Winding Properties** window. **RMxprt** automatically arranges the winding layout and display the relevant information that has been specified in the **Winding Editor** tab of the **RMxprt** main window. As long as the edited winding data have been saved, the **Winding Editor** tab will display the last saved winding data whenever **Winding Editor** dialog is launched. The left top part of the **Winding Editor** tab shows the winding data, as does the Winding Editor dialog box. In this area, the total number of rows equals half the number of slots.

Enabling the Winding Editor Dialog

Setting the **Winding Type** property to Editor enables the **Machine>Edit Layout** command on the menu bar. To display the **Winding Editor** dialog:

1. Open the **WindingProperties** window and set the **Winding** property to Editor.

   To do this, double-click on the Winding property button to display the **Winding Type** selection window.
2. Select **Editor** as the **Winding Type** and click OK.

   This closes the **Winding Type** window and sets the Winding Type property to Editor. It also enables the **Machine>Edit Layout** command on the menu bar.
3. Click **Machine>Edit Layout**.

   This displays the **Winding Editor** dialog. The **Winding Editor** dialog box includes functions that do not appear in the **RMxprt** main window **Winding Editor** tab.

   You can also invoke the Winding Editor dialog by:

   a. Right-click on the data table section of the Winding Editor tab of the main window.
   b. This displays an Edit Layout button.
   c. Click the Edit Layout button to display the Winding Editor dialog.

   You can also display the **Winding Editor** dialog by:

   a. Right click in the **Winding Editor** tab main window display.

This displays a shortcut menu.

b. Click **Edit Layout** from the shortcut menu.

Each row of the winding data is identified with coil index in the **Coil** column. This information is displayed in the **Winding Editor** tab in the RMxprt Main window, and it editable in the **Winding Editor** dialog.

- Column **Phase** is for the phase to which the coil belongs.
- Column **Turns** is for the number of turns of coil.
- Column **In Slots** is for the slot number with the coil side current flowing in ('flow-in-side' for short). If 2 Layers are specified in the Winding Properties window, the slot number ends with a "T" to show the top layer.
- Column **Out Slots** is for the slot number with the coil side current flowing out ('flow-out-side' for short). If 2 Layers are specified in the Winding Properties window, the slot number ends with a "B" to show the bottom layer.

By changing the belonging phase in column **Phase**, the number of turns in column **Turns**, the flow-in-side slot number in column **In Slot**, the flow-out-side slot number in column **Out Slot** for each coil, it is possible to arrange the distribution of coils of single and double layer winding of any type required.

The **Winding Editor** dialog also includes three check boxes:

- **Periodic Multiplier:** indicates the possibility to select the number of unit machines for editing winding arrangement. It has a drop-down menu to show the possible numbers for the periodic multiplier.

  When checked, the pull-down list box to the right displays the numbers of unit machines for selection. Selecting *1* means whole slots are considered as one unit machine, and all coils is listed in the table of the edit window. Selecting *2* lists half of the total coils in the table, and whole slots are divided into two unit machines, etc. When the check box **Periodic Multiplier:** is unchecked, the pull-down list box to the right is grayed (enabled); all the coils are listed in the table.

- Constant Turns.

  Checking the check box (multiple choices) **Constant Turns** indicates that the number of turns keeps constant and the column **Turns** in the table is grayed (disabled). If the check box **Constant Turns** is unchecked, the column **Turns** in the table is brightened allowing for editing and modifying the number of turns.

- Constant Pitch

  Checking this box grays the Out Slots column to the values cannot be edited. It means that the coil pitch is constant. For two-layer windings, all flow-in-side slots are defined as top layer, and all flow-out-side slots as bottom layer. The flow-out-side slot number is automatically computed based on the input in the edit box Coil Pitch in Stator2 page in RMxprt window, and Out Slot column is disabled.

  When the check box Constant Pitch is unchecked, the column Out Slot is enabled to allow arbitrarily changing slot pitch for each coil.

The **Winding Editor** dialog includes three command buttons.

- Click the command button **Default** in the window **Winding Editor**, all the data in the table resumes to the situation of data from automatic arrangement by **RMxprt**.
- Click the command button **Reset** in the window **Winding Editor**, all the data in the table resumes to the situation of data when the window **Winding Editor** was first opened, or resumes to the data that you have saved.
- Click OK to accept the current values and close the **Winding Editor** dialog.

Windings Basic Terminology

Conductor

A conductor refers to a half turn of a coil. A conductor may be formed with one insulated wire, or with several strands of insulated wires.

Strands

A conductor may consist of several wires of same or different sizes stranded together. The number of strands is also called **number of wires per conductor**. The conductor current may not uniformly distribute among all wires, but the current density is uniformly distributed.

Coil

A coil is wound with several **turns**, each turn consisting of two conductors. Coils are generally wound with insulation-wrapped electromagnetic wire continuously on a winding mould. However, coils with single-turn for heavy current are often formed with two separate thick conductors. A thick conductor is hammered onto the winding mould to form a half-coil. The linear part of a conductor imbedded into a slot of iron core is termed **effective side**.

Coil Pitch

The number of slots of the armature iron core spanned by the two effective sides of a coil is termed coil pitch, denoted by y. For instance, if the side of a coil in the 1-st slot spans 8 slots and is connected to the side of the coil in the 9-th slot, the coil pitch of the coil is y = 8.

Full coil pitch:	coil pitch = pole pitch
Short coil pitch:	coil pitch < pole pitch
Long coil pitch:	coil pitch > pole pitch, usually used in variable-pole multiple-speed machines
Pole pitch:	distance between two contiguous poles measured in number of slots. $polepitch = \frac{totalnumberofslots}{numberofpoles}$

Coil Set

The coils belonging to the same phase under one pole are connected in series as a coil set.

Winding

The coils or coil sets of a phase are connected according to certain rules to form a phase winding. A phase may consist of several **branches** connected in parallel. Every branch must produce exactly the same back emf and must have the same resistance. As a result, the phase winding current is uniformly distributed among all branches.

In summary, a **winding** may be connected with several branches in parallel; each **branch** consists of one or more coil sets connected in series; a **coil set** may have several series coils; a **coil** is wound with a number of turns; a **turn** is formed by two conductors; a **conductor** may be stranded by one or more same- or different-size wires.

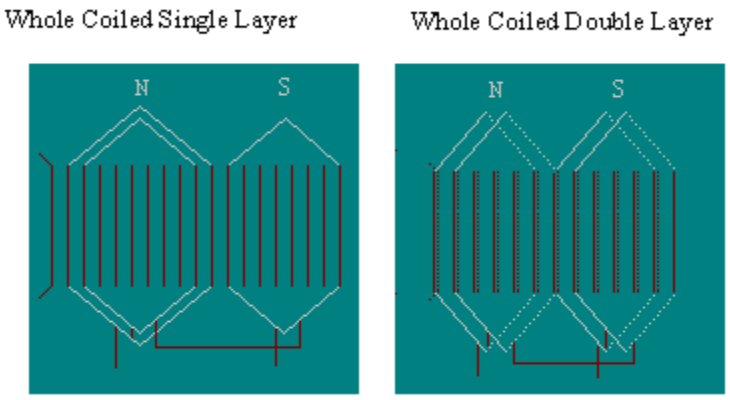
Poly Phase AC Winding

The common armature winding of poly-phase ac machines is catalogued and classified as shown in the following table.

Polyphase AC Winding	
Double layer	Variable-pole multiple speed type
	Fractional slot number type
	Wave-type
	Concentric type
	Lap Type
Single Layer	Crossed Concentric type (whole coiled or half coiled)
	Crossed Chain-type (whole coiled or half coiled)
	Concentric type (whole coiled or half coiled)
	Lap-type (whole coiled or half coiled)
	Chain-type (whole coiled or half coiled)
Compound layer	

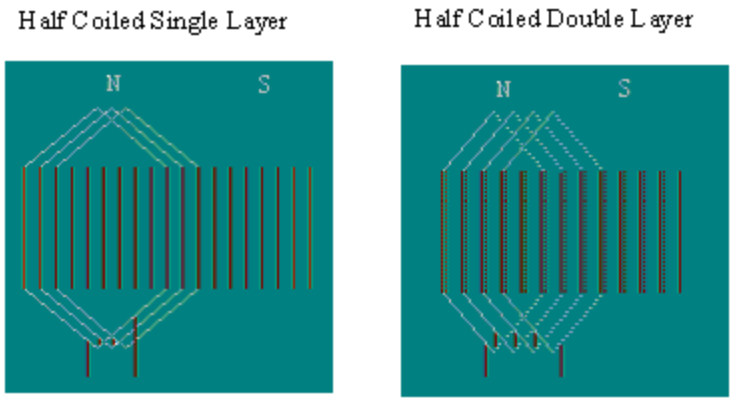
Whole-coiled Windings

When the coils of an AC winding are connected so that there are as many coil sets per phase as there are poles, the winding is called "whole-coiled."



Half-coiled Windings

When the coils are connected so that there is only one coil set per phase per pair of poles, the winding is called "half-coiled."



Single-Layer Windings

All the conductors in one slot are connected in series with all the conductors in another slot to form a single-layer coil. You set the number of winding layers in the Winding properties window, Winding tab. Comparing to double-layer type, this type is characterized by

- Number of coils halved;
- No need for insulation between layers, therefore higher slot filling factor;
- Coil pitch depends on the connection, and is not adjustable;
- Being widely used in small capacity electric machines.

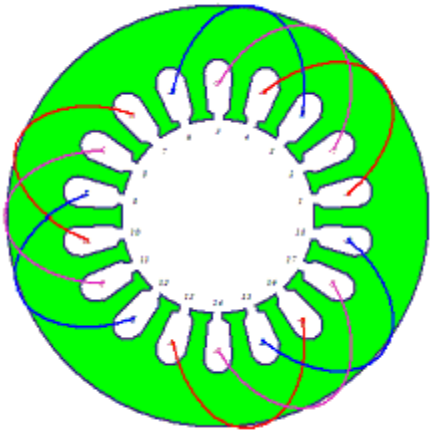
According to different layouts of the end winding, single-layer windings are classified as chain-, lap-, concentric- and crossed-types.

Chain-type Windings

The name single-layer chain-type is from the linked chain-like developed winding diagram. For a chain-type winding, every coil set has only one coil.

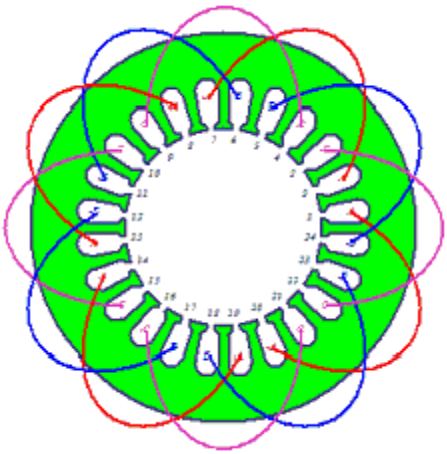
Half-coiled Chain-type Winding

An example of three-phase 6-pole 18-slot single-layer half-coiled chain-type winding is shown in the following figure.



Whole-coiled Chain-type Winding

An example of three-phase 4-pole 24-slot single-layer whole-coiled chain-type winding is shown in the following figure.

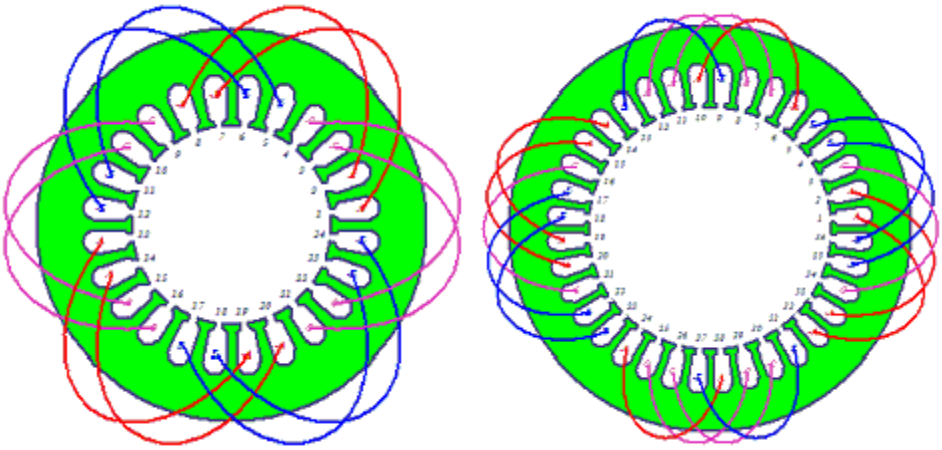


Lap-type Windings

The name single-layer lap-type is from the lapped layout of end connection. In a lap-type winding, at least one coil set has 2 or more coils which are overlapped each other. If some coil sets have only one coil, this winding type is also called "crossed lap-type".

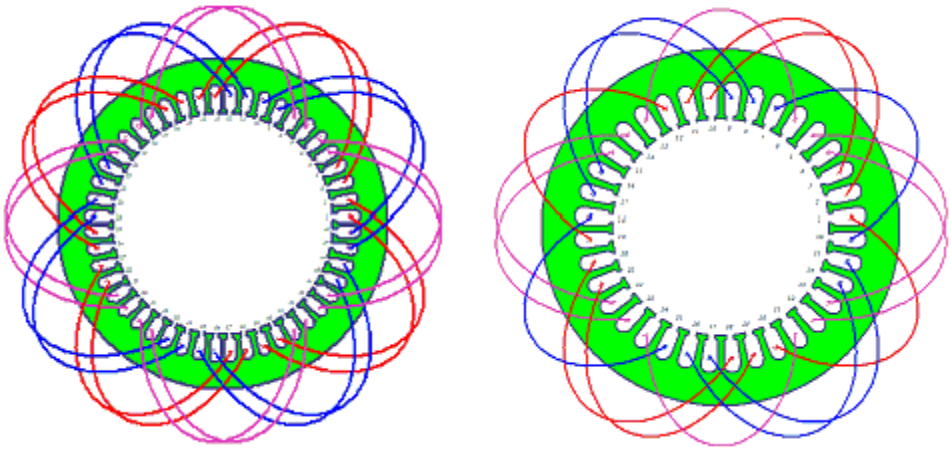
Half-coiled Lap-type Winding

An example of three-phase 4-pole 24-slot single-layer half-coiled lap-type winding is shown in the figure on the left, and an example of three-phase 8-pole 36-slot single-layer half-coiled crossed lap-type windings is shown in the following figure on the right.



Whole-coiled Lap-type Winding

An example of three-phase 4-pole 48-slot single-layer whole-coiled lap-type winding is shown on the left, and an example of three-phase 4-pole 36-slot single-layer whole-coiled crossed lap-type winding is shown on the right1



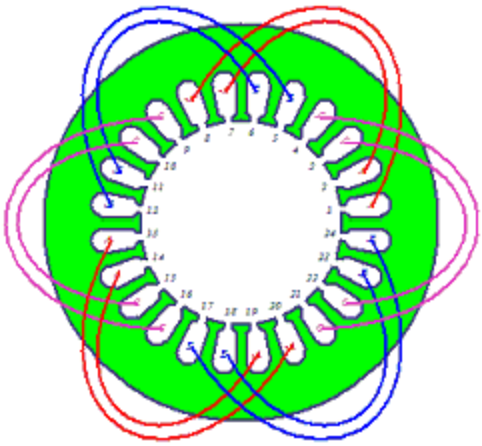
Concentric-type Windings

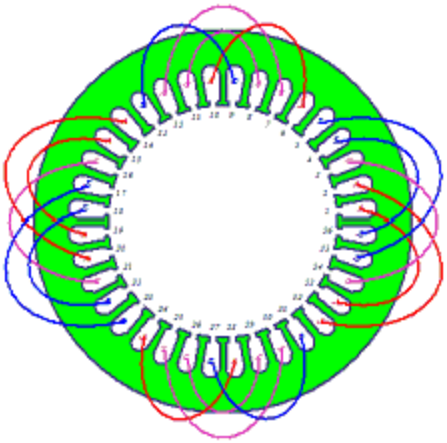
In a concentric-type winding, at least one coil set has 2 or more coils and non coils are overlapped each other. If some coil sets have only one coil, this winding type is also called "crossed concentric-type".

The single-layer concentric-type is formed of coils with different coil pitch, but with the same central line and of concentric-circle-like, therefore is named concentric-type. Its end connection can be arranged in layers, and therefore is convenient to imbed into slots. Nevertheless, the end magnetic leakage is a bit bigger.

Half-coiled Concentric-type Winding

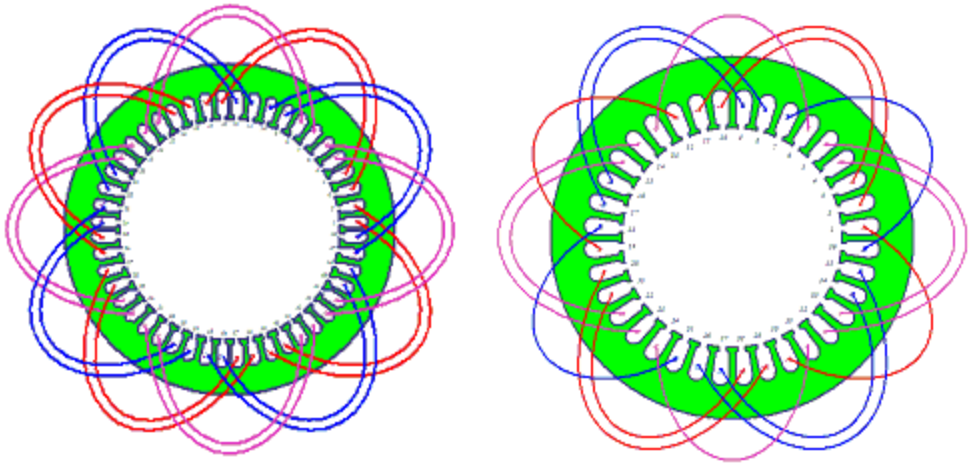
An example of three-phase 4-pole 24-slot single-layer half-coiled concentric-type winding is shown on the left, and an example of three-phase 8-pole 36-slot single-layer half-coiled crossed concentric-type winding is shown on the right.





Whole-coiled Concentric-type Winding

An example of three-phase 4-pole 48-slot single-layer whole-coiled concentric-type winding is shown on the left, and an example of three-phase 4-pole 36-slot single-layer whole-coiled crossed concentric-type winding is shown on the right.



Double-Layer Windings

In this type, the conductors in a slot are arranged in upper and lower layers. One side of each coil is imbedded in the upper layer in one slot and the other side is imbedded in the lower layer in another slot. You set the number of winding layers in the Winding properties window, Winding tab. Comparing to single-layer-type, this type is characterized by:

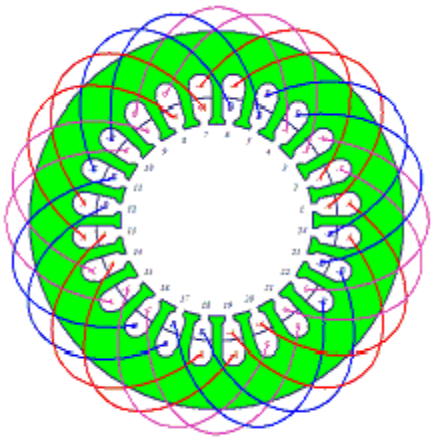
- Number of coils doubled;
- Need for insulation between layers, therefore lower slot filling factor, and danger in electric breakdown between phases;
- Adjustable coil pitch, therefore possible weakening of harmonic emfs with proper short pitch factor to improve electromagnetic properties of electric machines;
- Being widely used in electric machines with capacity over 10 kW.

For the single speed electric machine, the double-layer winding typically adopts whole-coiled type. For the double speed electric machine with doubling number of poles, the double-layer winding is whole-coiled in high speed, half-coiled in low speed.

According to different coil shapes, double-layer windings are classified as lap-, concentric- and wave-types.

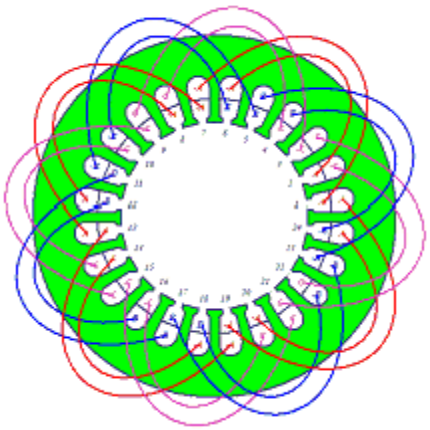
Double-layer Lap-type Winding

An example of three-phase 4-pole 24-slot whole-coiled double-layer lap-type windings (short pitch $y = 5$) is shown below.



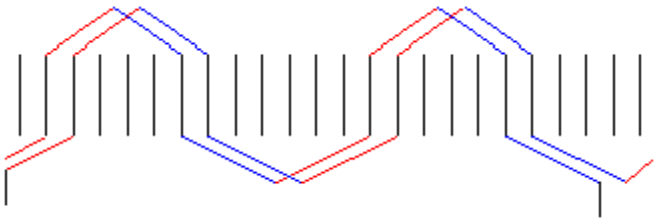
Double-layer Concentric-type Winding

An example of three-phase 4-pole 24-slot whole-coiled double-layer concentric-type windings (short pitch $y = 5$) is shown below.



Double-layer Wave-type Winding

The name double-layer wave-type is from the wave-like developed winding diagram as shown below.



Compared to lap-type:

- The winding of each phase connects the coils under different poles in series in one round, and returns to the left to the first coil, then winds the next round, and so on so forth until all the coils belonging to this phase are connected.
- This type is usually used in single-turn preformed hard coil for low voltage high current electric machines.
- This type needs less connection wire between poles.

Fractional-Pitch Winding

First, introduce a number *q*, called **number of slots per pole per phase**, which is defined as

$$q = \frac{total\,number\ of\ slots}{number\ of\ poles \times number\ of\ phases}$$

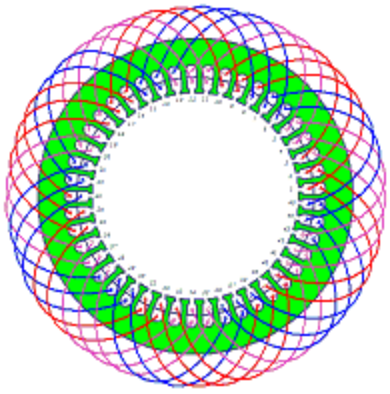
$$q = b\frac{c}{d}$$

A fractional-pitch winding has a fractional number .

An example of three-phase 6-pole 45-slot fractional-pitch double-layer winding

$q = 2\frac{1}{2}$ $\tau = 7\frac{1}{2}$

(, short pitch *y* = 7, pole pitch) is shown here.



Auto-arrangement of AC Windings

RMxprt can arrange these windings automatically if all coils have the same number of turns. This section describes the process to automatically arrange the coil distribution.For winding layout display in **RMxprt**, the lap-type is default if windings are automatically arranged. If a concentric-

type layout display is desired, the winding can be defined by winding editor, as described in the next section.

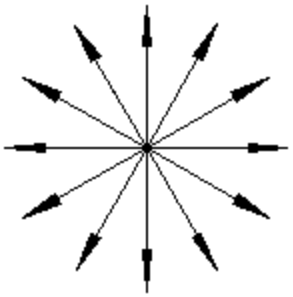
The wave-type winding is effective to a lap-type winding, and is also displayed as a lap-type winding.

Star Vector Diagram

The conductors (or coils) in slots produce emf (or mmf), which can be expressed with unit vector. When the electric machine has number of pole *p*, and number of slots *Z*, the angular phase difference in electric degrees between two contiguous slots is

$$\alpha = \frac{p \times 180^{\circ}}{Z}$$

Drawing the vectors of emfs (or mmfs) in all the slots according to their phase angles forms the star vector diagram of the winding. The following figure shows an example of the star vector diagram of 4-pole 24-slot winding.



If there exists the greatest common factor *t* between the number of slots *Z* and the number of pole pairs *pp* (= *p/2*), the star vector diagram repeats *t* times, i.e. the winding has *t* periods. Let

$$Z_0 = \frac{Z}{t}$$

,

and

$$p_0 = \frac{p}{t}$$

,
then Z_0 and p_0 construct a complete star vector diagram and form a unit electric machine. For the whole-pitch winding electric machine (q, as shown later, is an integer), $t = p/2$. For the fractional-pitch winding electric machine,

$$q = \frac{Z}{mp} = b + \frac{c}{d}$$

where m is the number of phases. If $t > 1$, the angular phase difference between two contiguous vectors is

$$\alpha = \frac{360°}{Z_0}$$

and the difference between the ordinal numbers of the slots of two contiguous vectors is

$$y_0 = \frac{m(bd + c)G - 1}{d}$$

where G is a minimum integer to make y_0 equal to an integer (y_0 should take into account the possible reverse connection of coils under the contiguous pole).

Phase Spread

In the star vector diagram of a unit electric machine, the range occupied by the vectors of each phase under one pole is termed phase spread, expressed in electric degrees or number of slots. For a single-layer winding, the phase spread is 180°/m (m – the number of phases). The phase spread of a double-layer half-coiled winding is 360°/m, and the phase spread of a double-layer full-coiled winding is 180°/m.

The phase spread of a 2-phase winding is always 90° (= 180°/m). Therefore, a 2-phase winding cannot take the double-layer half-coiled winding type. The windings for single-phase induction motor are also considered as 2-phase windings.

When the number of phases is an even number of greater than or equal to 4, the phase spread is always 360°/m. Therefore, a winding with even number of phases (4, 6, …) can take only the double-layer half-coiled winding type.

When the number of phases is an odd number of greater than or equal to 3, the phase spread can be either 360°/m or 180°/m. Therefore, a winding with odd number of phases (3, 5, …) can take any winding types.

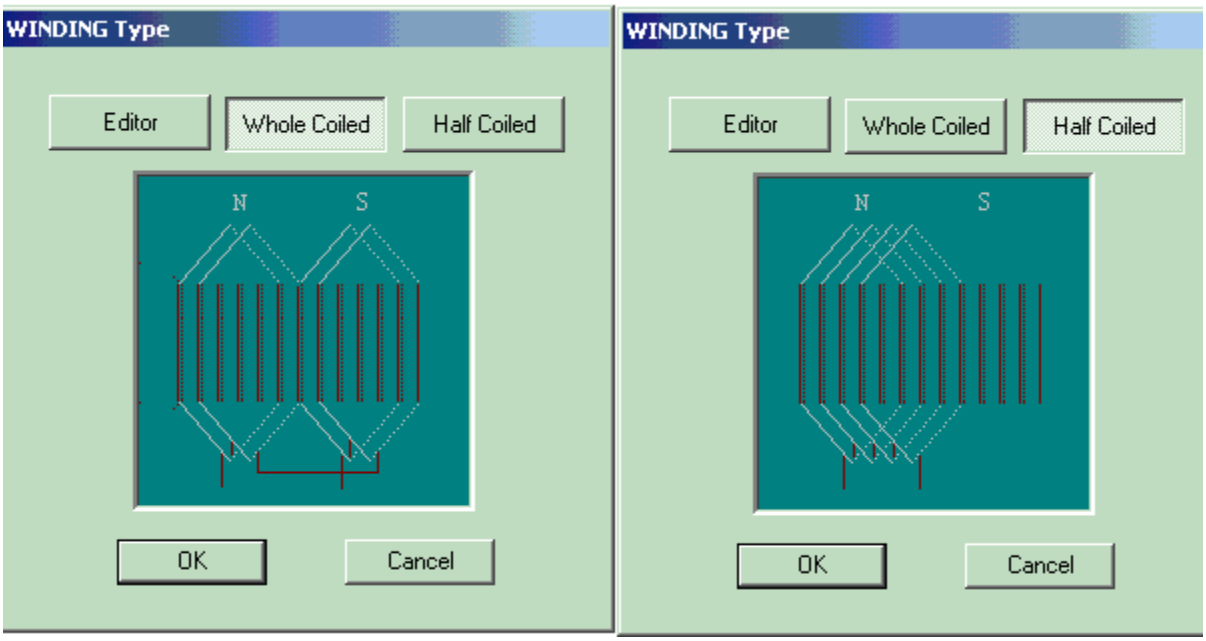
Coil Arrangement

Coil arrangement is completed by the following processes. First, draw the star vector diagram based on number of slots and number of poles. Then divide the whole region (360 electric degrees) to several phase spreads, which is derived from the number of phases and the winding type. Finally, assign all phase spreads to each phase in such a way that the axis of each succeeded phase lags by 360/*m* electric degrees (90 electric degrees for 2 phases).

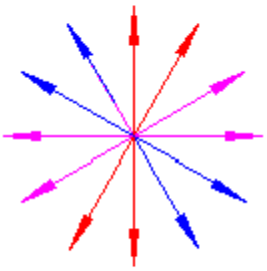
Double-layer Windings

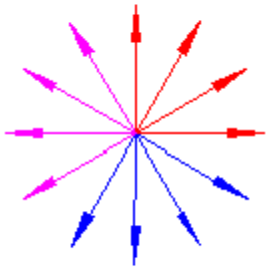
Take a three-phase winding as an example. The width of phase spread of half-coiled winding is $360^o / 3 = 120^o$, the sequence of the phase spread is A, B, C. For whole-coiled winding, the width of phase spread is $180^o / 3 = 60^o$, the sequence of phase spread is A, –C, B, –A, C, –B, where the phase spread with negative sign is termed negative phase spread.

The winding types can be set in the Winding Type panel for a machine that includes these options (in this case, a brushless permanent magnetic DC motor), for double-layer whole-coiled windings as shown in on the left and double-layer half-coiled windings as shown on the right.



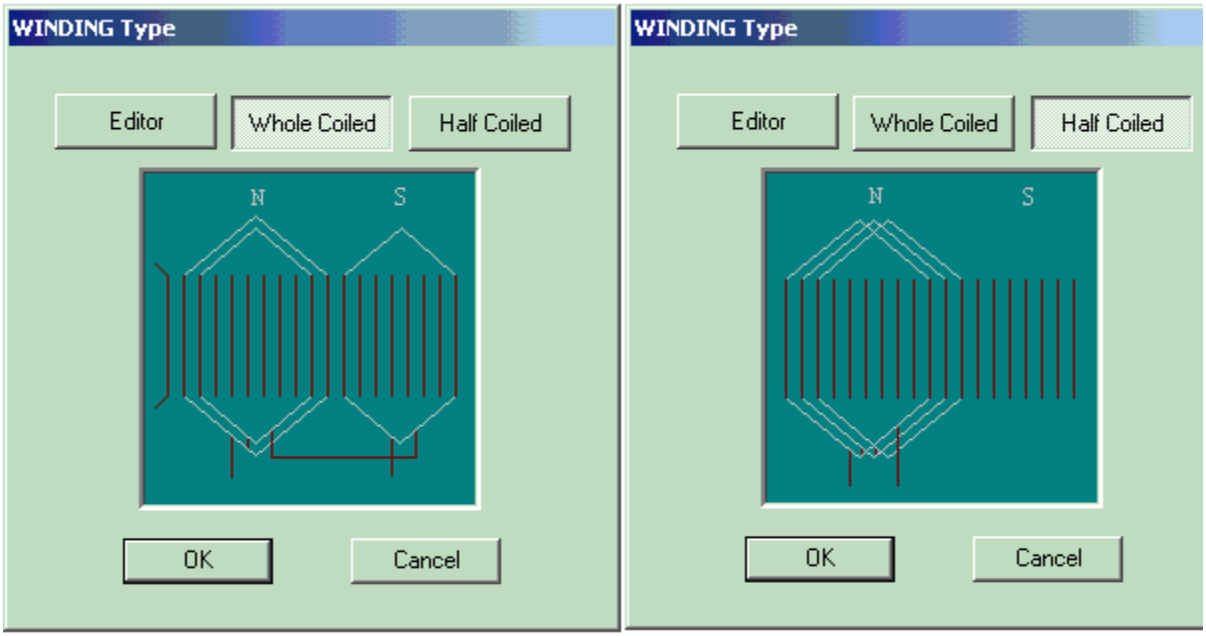
The star vector diagram of a three-phase whole-coiled (60^o-phase-spread) winding is shown below on the left, and that of a half-coiled (120^o phase spread) winding is shown below on the right.





Single-layer Windings

The winding layers can be set in the properties window for the winding, for single-layer whole-coiled windings as shown on the left and single-layer half-coiled windings as shown on the right.



The phase spread of a three-phase single-layer whole-coiled or half-coiled winding is 60^{o}, and the star vector diagram is the same as the double-layer whole-coiled winding.

Fractional-pitch Windings

The number of slots per pole per phase of fractional-pitch winding is a mixed number.

$$q = b\frac{c}{d}$$

In the unit electric machine, the numbers of slots occupied by phase spread are not all the same, but repeat with the radix *d*. In each *d* poles, there are *c* poles with the slot number of phase spread equal to *b* + 1 (big phase spread), *d* – *c* poles with the slot number of phase spread equal to *b* (small phase spread).

Take as an example a three-phase 10-pole 36-slot fractional-pitch winding with phase spread of 60°. The number of slots per pole per phase of fractional-pitch winding is

$$q = \frac{36}{3 \times 10} = 1\frac{1}{5}$$

the greatest common factor between the number of slots 36 and the number of pole pairs 5 is *t* = 1, the angular phase difference between two contiguous vectors in the star vector diagram is

$$\alpha = \frac{360^{\circ}}{36} = 10^{\circ}$$

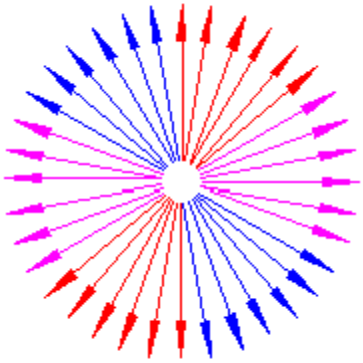
the difference between the ordinal numbers of the slots of two contiguous vectors is (*G* = 2)

$$y_0 = \frac{3(1 \times 5 + 1) \times 2 - 1}{5} = 7$$

the repetition radix *d* = 5. In each 5 pole region, each phase has big phase spread of 1 + 1 = 2 slots under 1 pole, and small phase spread of 1 slot under 4 poles. The repeating format is 2 1 1 1 1 for phase A. The repetition of phase spread distribution for all phases is shown in the following table.

Slot number	1~2	3	4	5	6	7~8	9	10	11	12	13~14	15	16	17	18
Phase spread	A	–C	B	–A	C	–B	A	–C	B	–A	C	–B	A	–C	B
Slot number	19~20	21	22	23	24	25	27	28	29	30	31	33	34	35	36
Phase spread	–A	C	–B	A	–C	B	–A	C	–B	A	–C	B	–A	C	–B

The star vector diagram of winding is shown below.



Asymmetric Windings

Whole-pitch windings (*q* is integer) are always symmetric. Fractional-pitch windings with

$$q = b\frac{c}{d}$$

becomes asymmetric if the denominator *d* is a multiple of the number of phases *m*. In general, it is avoid using asymmetric windings as possible. Nevertheless, it is sometime possible to design poly-phase windings with little asymmetry in order to use existing punching tools.

If *d* is a multiple of the number of phases *m*, but the total number of slots *Z* can be divided by *m*, it is possible to construct poly-phase winding with little asymmetry. **RMxprt** can perform automatic arrangement for this sort of windings and obtain the phase-spread in electric degrees for each phase.

Take as an example a three-phase 6-pole 66-slot fractional-pitch winding electric machine. Since

$$q = \frac{66}{3 \times 6} = 3\frac{2}{3}$$

d = *m* = 3, the winding is asymmetric. The output in the window **Design Output** is shown below.

The information for WINDING ARRANGEMENT is displayed as follows:

The distribution of coil slots to phases:

The 3-phase, 2-layer winding can be arranged in 66 slots as below:

AAAAZZZZBBBXXXXCCCCYYYAAAZZZZBBBBXXXCCCCYYYYAAAAZZZBBBBXXXX CCCYYYY

X, Y and Z stands for –A, -B and –C, respectively. For asymmetric windings, additional information is output, as shown below.

The winding factors of each phase are:

Phase A	0.954119
Phase B	0.954119
Phase C	0.949042

The angles between two-phase winding axes are:

Phase A & B	119.082
Phase B & C	120.459
Phase C & A	120.459

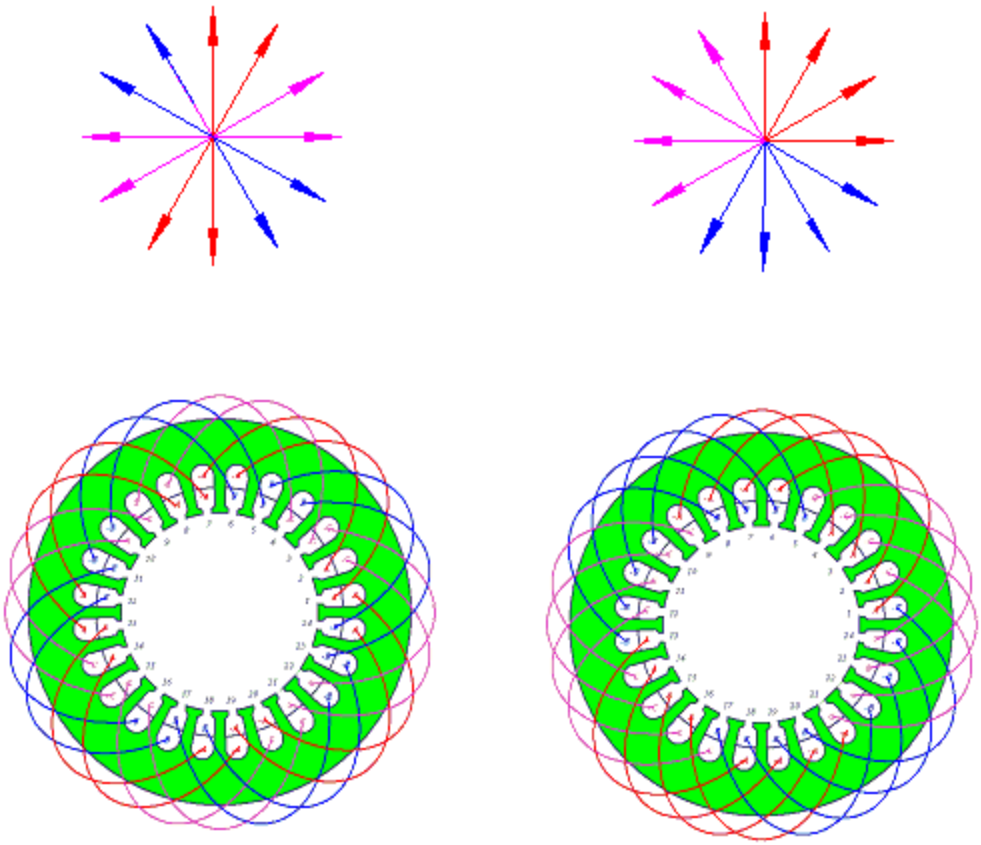
If a sinusoidal rotating field links the winding, the fundamental induced-voltage components will be:

Positive-sequence component	100%
Negative-sequence component	0.286577%
Zero-sequence component	0.639823%

Coil Connections

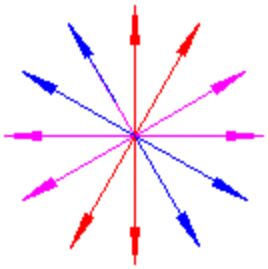
Connection of Double-layer Lap Windings

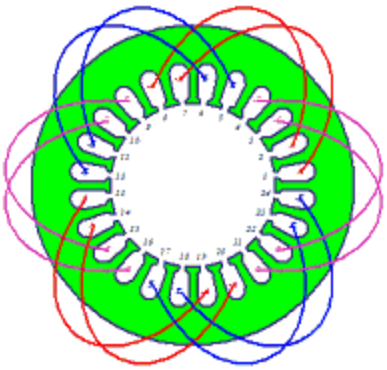
Every vector represents the top-layer effective side of a coil. The bottom effective side of the coil is determined based on the coil pitch, and is not displayed in the diagrams. Therefore, every vector in the diagrams can also stand for a coil. Connect all coils in phase spread of A in positive direction, and all coils in phase spread of –A in negative direction to form the phase A winding. In this way, phase B and C windings can also be connected. The winding connection layouts for the vector diagrams are shown below.

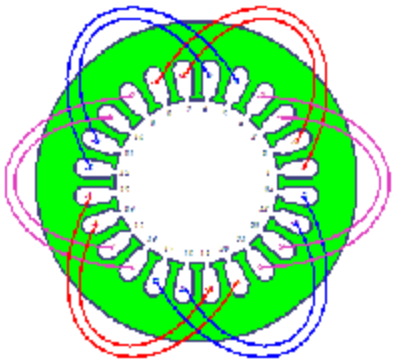


Connection of Single-layer Half-coiled Windings

Every vector in A, B and C phase spread represents "go" effective side of a coil, the "return" effective side of the coil is located in –A, –B and –C phase spread. For the lap type connection, all coils are with full coil pitch. The connection layouts of the lap type and the concentric type, with respect to the same vector drawing are shown below.



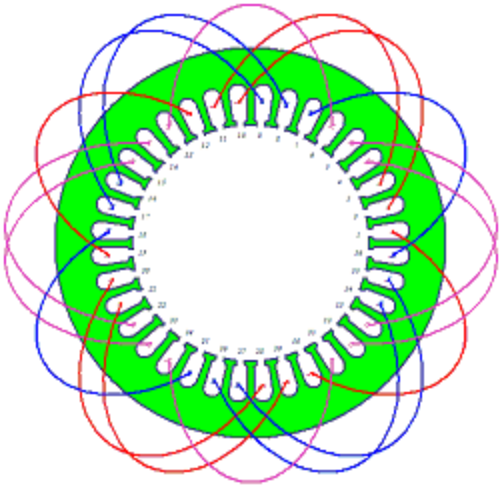




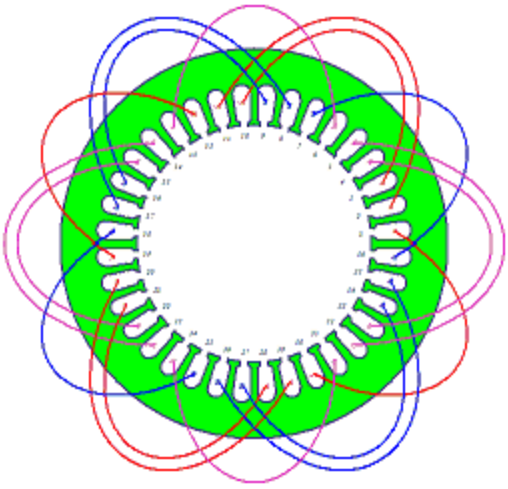
Connection of Single-layer Whole-coiled Windings

In the previous example, for the concentric type (lower right in the diagram), if coil 1 is not connected from slot 1 to slot 8 (long coil pitch: coil pitch = 7 > pole pitch = 6), but connected from slot 1 to slot 20, and slot 8 to slot 13, all coils of phase A winding have coil pitch of 5. In this way, the winding becomes single-layer whole-coiled type with the same star vector diagram and phase spread, and has much shorter average coil pitch. Therefore, single-layer whole-coiled windings consume less electromagnetic wire than single-layer half-coiled windings. **RMxprt** can optimize connections to minimize the average coil pitch to form a single-layer whole-coiled winding.

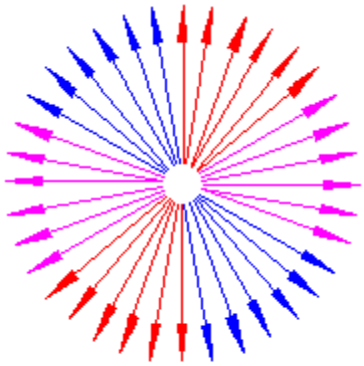
An example of three-phase 4-pole 36-slot single-layer whole-coiled crossed lap-type winding (q = 3, 60^o phase-spread) is shown below.



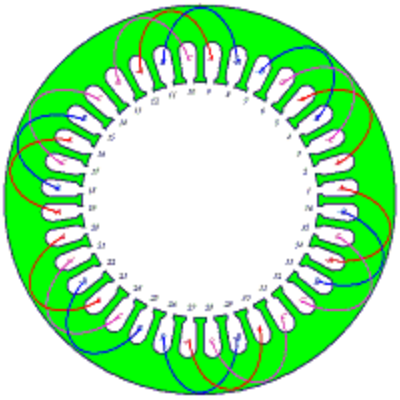
An example of three-phase 4-pole 36-slot single-layer whole-coiled crossed concentric-type winding (q = 3, 60^o phase-spread) is shown below.



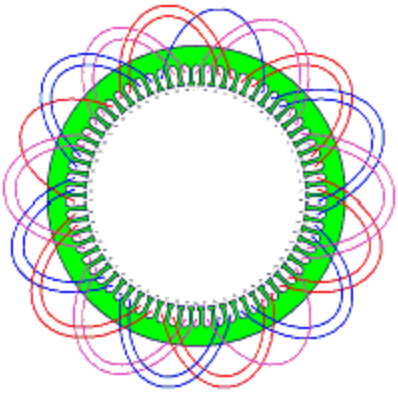
A star vector diagram with fractional coil pitch can also be connected with single-layer whole-coiled type. When the number of slots per pole per phase q <2, as shown in the following vector diagram, the number of coil sets per phase may not equal to the number of poles (6 coils vs 10 poles), but the algorithm to connect coils is the same (minimize the average coil pitch), and therefore, it is still referred as whole coiled windings.



The winding connection layout for the previous vector diagram is shown below.



Another example is an asymmetric three-phase winding. The connection layout is shown below.



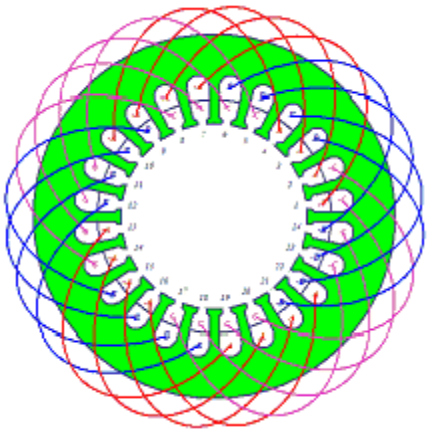
Connection of Double-pole Dual-speed Windings

Commonly used for coils of double-layer windings are lap- and concentric-types. Single-speed double-layer winding electric machine is usually connected as whole-coiled winding. Double-pole dual-speed electric machine is connected as double-layer whole-coiled winding at high speed and as double-layer half-coiled winding at low speed.

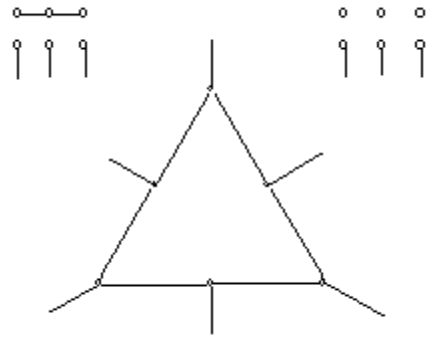
Take as an example a three-phase dual-speed 2/4-pole 24-slot double-layer winding. The arrangement of coils is shown in the Table and in the following figure.

Slot number	1	2	3	4	5	6	7	8	9	10	11	12
Two poles	A	A	A	A	–C	–C	–C	–C	B	B	B	B
Four poles	A	A	A	A	C	C	C	C	B	B	B	B

Slot number	13	14	15	16	17	18	19	20	21	22	23	24
Two poles	–A	–A	–A	–A	C	C	C	C	–B	–B	–B	–B
Four poles	A	A	A	A	C	C	C	C	B	B	B	B



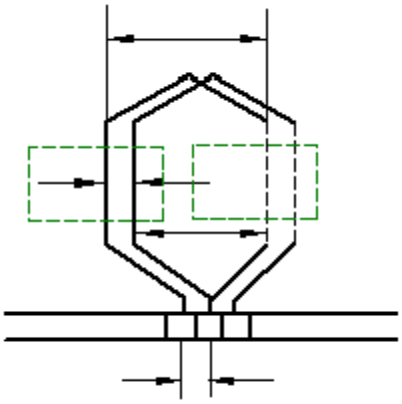
The connection for variation of number of poles is shown in the next figure. On the top-left is the 2-pole 2Y-connection at high speed, on the top-right is the 4-pole -connection at low speed.



DC Windings

Lap Winding

The winding layout of a lap-type winding for dc machines is as shown below.



A lap-type winding has the following relationships:

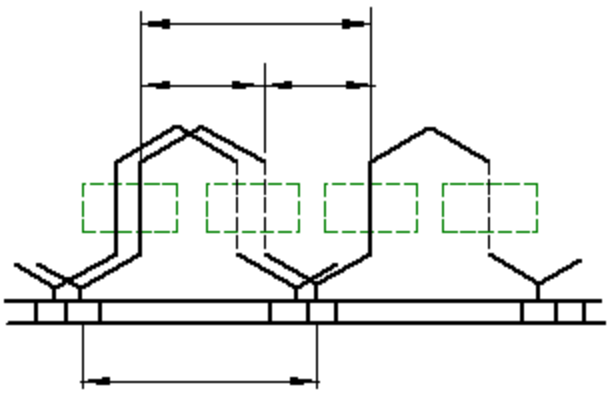
Coil pitch	$y_1 = \frac{Z}{p} \mp \varepsilon = \text{integer}$
Commutator pitch	$y_k = \pm m$
Number of branches in parallel	$a = mp$

where

Z	number of slots
p	number of poles
m	number of multiplex

Wave Winding

The winding layout of a wave-type winding for dc machines is as shown below.

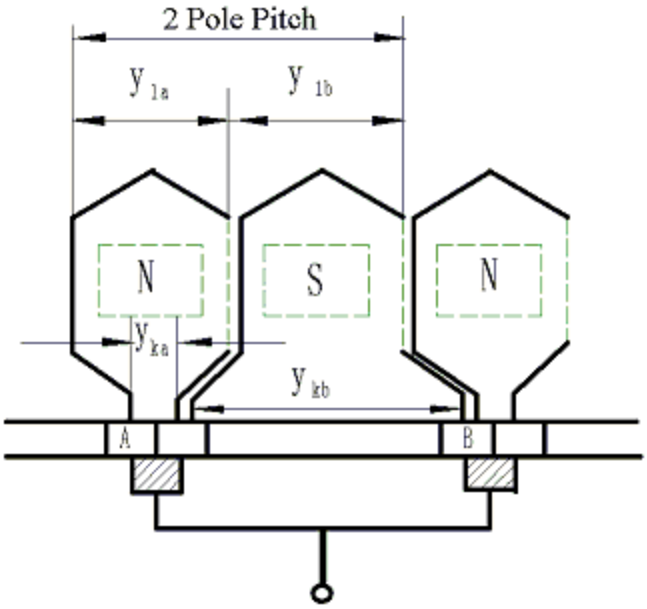


A wave-type winding has the following relationships:

Coil pitch	$y_1 = \frac{Z}{p} \mp \varepsilon = \text{integer}$
Commutator pitch	$y_K = \frac{K \mp m}{p/2} = \text{integer}$
Number of branches in parallel	$a = m$

Frog-leg Winding

A frog-leg winding consists of a lap winding and a wave winding. Assume the lap winding has *m* multiplex number. In order to connect the wave winding in parallel with the lap winding, the wave winding should have the same branch bake emf as the lap winding. Therefore, the number of branches in parallel of the wave winding has to be the same as that of the lap winding, or the multiplex number of the wave winding must be *mp/2*. The winding layout of a frog-leg-type winding with *m=1* for dc machines is as shown below.



A frog-leg-type winding has the following relationships:

Coil pitch	$y_{1a} + y_{1b} = \frac{2K}{p} = \text{integer}$
Commutator pitch	$y_{Ka} + y_{Kb} = \frac{2K}{p} = \text{integer}$ $y_{Ka} = \pm m$

	$y_{k2} = \frac{2K}{p} \mp m$
Number of branch in parallel	$a = pm$

where

m	number of multiplex (of the lap winding)

Virtual Slots

Windings of the dc machine are usually double-layer type. In many cases, often a number of coil sides are imbedded into one slot for simplification of structure. Therefore, the number of slots Z is less than the number of coils S. There exists the relationship

$$Z = \frac{S}{\mu}$$

where μ is the number of coil sides in each layer in one slot and is termed virtual slot factor. Therefore, the number of conductors per slots is equal to multiple of 2μ.

Equipotential Connectors

The points ideally possessing the same electric potential in armature winding are often wholly or partly connected by short copper wire, which is termed equipotential connector.

Equipotential Connector Class A of Simplex-lap Winding

Asymmetry in magnetic circuit, such as eccentric air-gap, causes circulating current in lap winding, increases losses and affects commutation in order. The equipotential connector Class A on the commutator of simplex-lap winding can solve this problem. The connection pitch of equipotential connector Class A of simplex-lap winding y_p is equal to the number of commutator segments per pole pair.

$$y_p = \frac{K}{p} = \frac{K}{a} = \text{integer}$$

Winding with y_p = integer is termed symmetric winding. Only symmetric winding can have equipotential connector Class A.

No need for Equipotential Connector for Simplex-wave Winding

Simplex-wave winding does not possess electrically equipotential points, therefore, can not have equipotential connector. On the other hand, simplex-wave winding does not need equipotential connector.

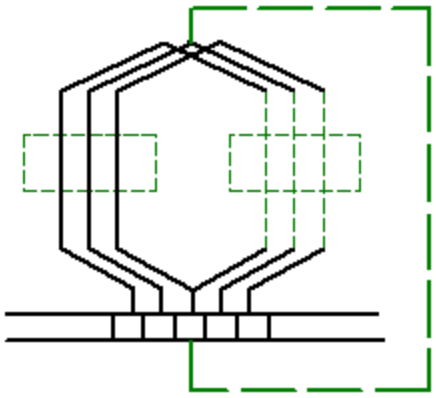
Equipotential Connector Class B of Multiplex-wave Winding

There are electric equipotential points among different sets of simplex-wave windings of multiplex-wave winding. They can be connected to eliminate the nonuniform distribution of electric potential on commutator segments due to inequality of brush resistances. This is termed equipotential connector Class B. The connection pitch of equipotential connector Class B of multiplex-wave winding y_p is equal to the number of commutator segments per branch pair in parallel.

$$y_p = \frac{K}{a} = \text{integer}$$

Equipotential Connector of Multiplex-lap Winding

For multiplex-lap winding, equipotential connector Class A is needed for each set of simplex-lap windings, equipotential connector Class B is needed among different sets of simplex-lap windings. There exist no electrically equipotential points among different sets of simplex-lap windings on the commutator side of armature, however, there exist electrically equipotential points among different sets of simplex-lap windings on different sides of armature as points A and B in the following figure.



Those points can be connected by conductors passing through inside armature.

No Need for Equipotential Connector for Frog-leg Winding

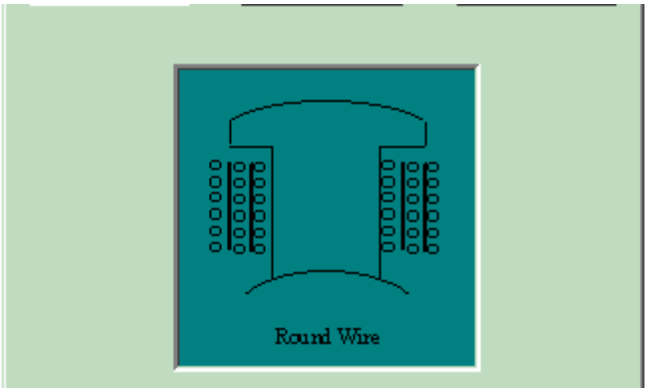
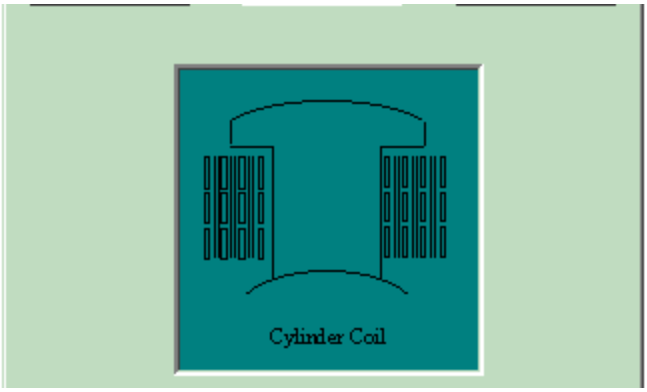
For frog-leg winding, each pair of electrically equipotential points on commutator is connected by a lap coil and a wave coil in series. The connection acts as equipotential connector Class A for lap winding and as equipotential connector Class B for wave winding. Therefore there is no need for extra equipotential connectors.

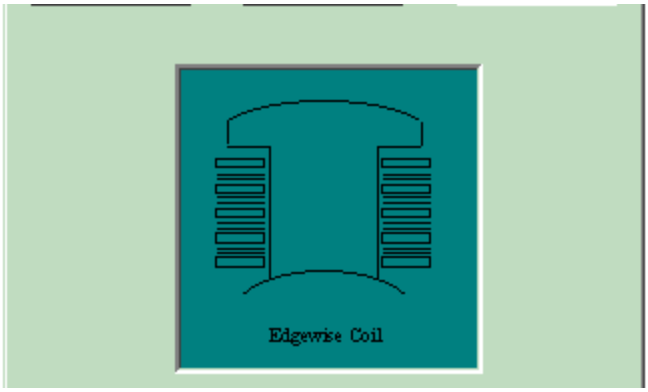
Pole Windings

The following two types of electric machines possess similar pole winding structure:

- DC machine (motor and generator)
- Three-phase synchronous machine (motor and generator)

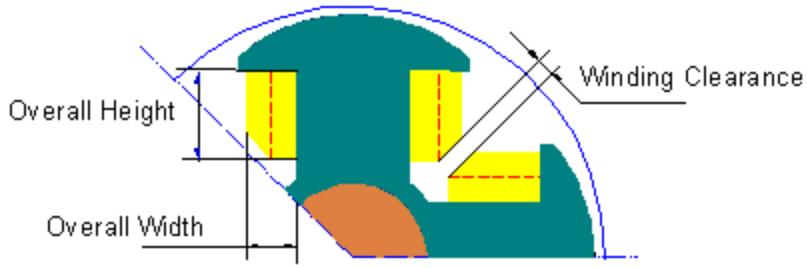
RMxprt adopts the same arrangement procedure for pole windings of the two types of electric machines. There are three types of structure as shown in the following figure:

Round Wire	pole winding with round wire. 
Cylinder Coil	pole winding with rectangular wire wound in standing way. 
Edgewise Coil	pole winding with rectangular wire wound in flatting way.

Limited Space for Wire Arrangement

Before completing a winding arrangement, **RMxprt** needs to determine the limited space sizes for the winding. The limited sizes include: limited **Overall Height**, limited **Overall Width** and **Winding Clearance** (the clearance between two adjacent pole windings), as shown below.

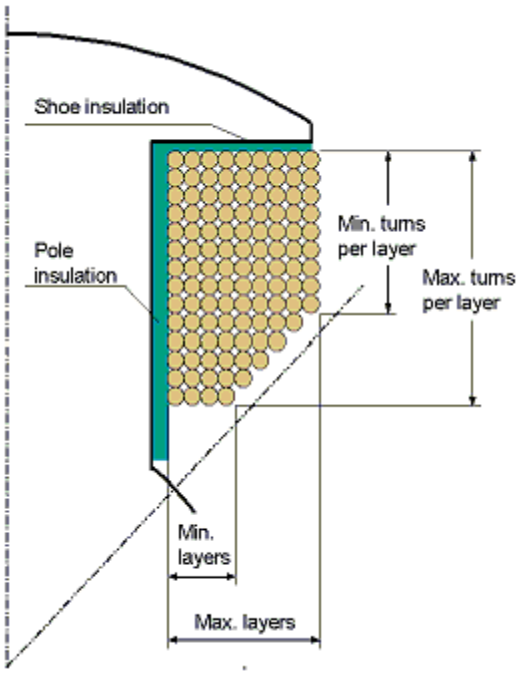


Overall Height, **Overall Width** and **Winding Clearance** can be input from RMxprt panel. If either **Overall Height** or **Overall Width** is set to *0*, **RMxprt** perform automatic space optimization to obtain the maximum space for pole winding arrangement in the condition to guarantee the clearance between pole windings.

If the space from input or determined by space optimization is not sufficient to arrange for the input number of turns per pole, a message of "The rotor/shunt/series/commutating winding control dimension is not big enough." is displayed in **Design Output** window.

Round Wire Winding

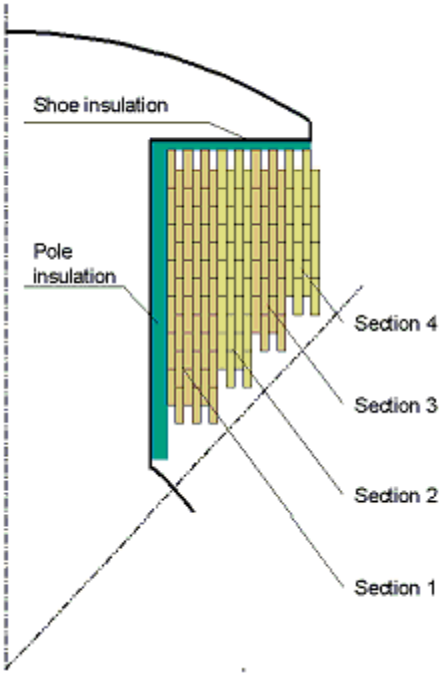
The arrangement of pole winding with round wire is shown below.



With auto-arrangement of pole windings, **RMxprt** calculates maximum number of layers, minimum number of layers, maximum number of turns per layer, minimum number of turns per layer, and the maximum number of turns per pole that is available from the limited space, and so forth.

Cylinder Coil

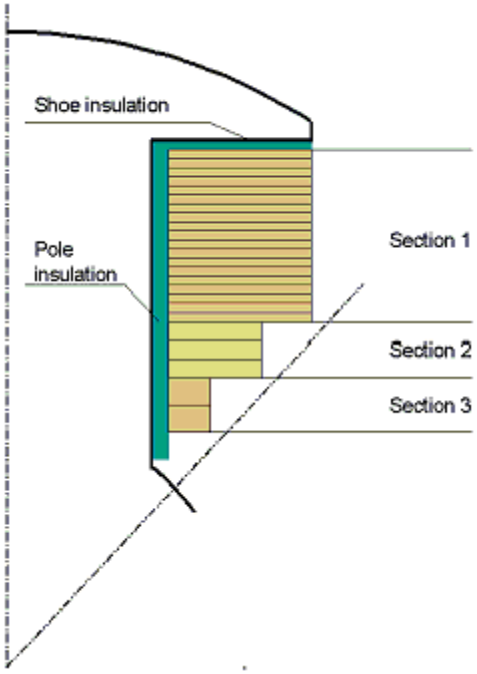
The arrangement of magnetic-pole winding with rectangular wire wound in standing way by **RMxprt** is shown below.



The cylinder coil is wound with half-turn over lapped layer by layer. Layers with the same number of turns constitute a section. The output window **Design Output** displays the number of layers and the number of turns per layer of each section, and the maximum number of turns per pole that is available from the limited space.

Edgewise Coil

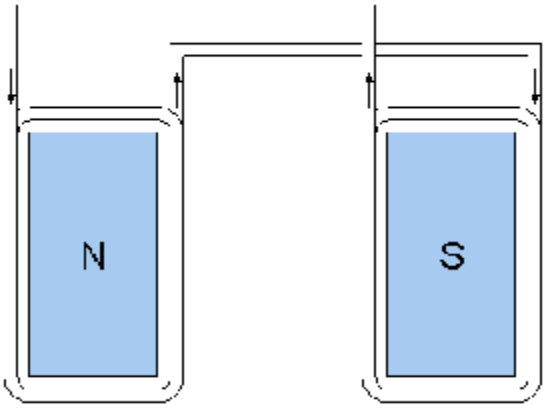
The arrangement of magnetic-pole winding with rectangular wire wound in flatting way by **RMxprt** is shown below.



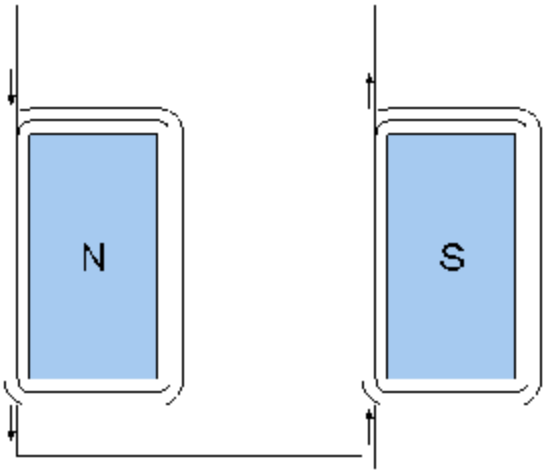
To guarantee the clearance between two adjacent pole windings, the wire width of the lower parts is decreased, while the wire thickness is increased to keep the sectional area of wire invariant as possible. The turns with the same wire gauge constitute a section (maximum three sections are allowed). The output window **Design Output** displays the number of turns of each section and the sizes of wire gauge, and the maximum number of turns per pole that is available from the limited space.

Pole Winding with Half Turns

For some large machines, a half turn may be included due to too few turns per pole. When the number of turns per pole is an integer number, the number of conductors per pole is an even number with equal conductor number at both pole sides. Therefore, the two terminal leads of one pole coil are at the same axial side (or at different pole sides), as shown below.



When the number of turns per pole includes a half turn, the number of conductors per pole is an odd number. In this case, one pole side has one more conductor than the other pole side, and the two terminal leads of one pole coil are at different axial sides (or at the same pole side), as shown below.



Exporting Winding Data

Winding data may be export to a table:

1. Select the **Machine>Winding>Export Layout** command from the menu. You may also right-click in the **Winding Editor** window and select **Export Layout** from the shortcut menu.

2. Browse to the location to save the file and enter a filename.
3. Click **Save** to export the winding data to a file and dismiss the dialog.

Note	The winding must be editable for the Export Layout command to be available. If you are using a standard winding, you can switch to the Winding Editor by: 1. Click on the **winding** in the Project Tree window. 2. In the Properties Window, click on the button next to **Winding Type**. The **Winding Type** dialog is displayed. 3. Select **Editor** as the winding type and click **OK**.

RMxprt Machine Types

Using Ansys RMxprt, you can simulate and analyze the following standard machine types:

- Adjust-Speed Synchronous Machine
- Brushless Permanent-Magnet DC Motor
- Claw-Pole Alternator
- General DC Machine
- Generic Rotating Machine
- Line-Start Permanent-Magnet Synchronous Motor
- Permanent-Magnet DC Motor
- Single-Phase Induction Motor
- Switched Reluctance Motor
- Three-Phase Induction Motor
- Three-Phase Non-Salient Synchronous Machine
- Three-Phase Synchronous Machine
- Universal Motor

Three-Phase Induction Motors

After you have selected **Three-Phase Induction Motors** as your model type, you must define the following:

- General data, such as the voltage, speed, and materials.
- Stator data, such as the slot types, diameter, and wire dimensions.
- Rotor data, such as the slot dimensions, skew, and ventage holes.
- Solution data, such as rated output voltage and frequency.

By option, you can:

- Add a machine housing.
- Add vents to and remove an existing vent from the stator and rotor.

Analysis Approach for Three-Phase Induction Motors

For a three-phase induction motor, the stator winding (with a sinusoidal spatial distribution and *p* pairs of poles) is connected to a three-phase symmetric voltage power supply. The resulting currents in the stator produce a rotating magnetic field. The rotor winding is often a squirrel cage type with the number of poles dictated by the number of poles in the stator. Currents are induced in the rotor bars and produce, in turn, a second rotating magnetic field. The two rotating fields produce a resultant rotating magnetic field in the air gap of the machine. The interaction of this field in the air gap with the rotor bar currents produces an electromagnetic torque, which acts on the rotor in the direction of the rotation of the field in the air gap. A torque of equal value acts upon the stator in the opposite direction.

The stator winding, which is connected to a phase of the supply system, has *p* coils, each with a symmetric spatial distribution and an opening of $\pi D/2p$, where *D* is the diameter of the winding. In this case, the magnetic field in the air gap has *p* periods, and the winding has *p* pairs of poles.

The performance of three-phase induction motors (IndM3) is analyzed based on the equivalent circuit of one phase in the frequency domain as shown in Figure 1.

In the figure, R1 is the stator resistance, X1 is stator leakage reactance, which consists of stator slot leakage reactance, end-winding leakage reactance, and differential leakage reactance. X2 and R2 are rotor leakage reactance and rotor resistance, respectively. X2 includes rotor slot leakage reactance, end-ring leakage reactance, differential leakage reactance, and skewing leakage reactance. Due to the saturation of the leakage field, X1 and X2 are nonlinear. The parameters in the equivalent circuit are dependent on the stator and rotor currents. Due to the skin effects, X2 and R2 are the equivalent values from a distributed-parameter circuit, as shown in Figure 2.

They vary with the rotor slip s. All rotor parameters have been referred to the stator side.

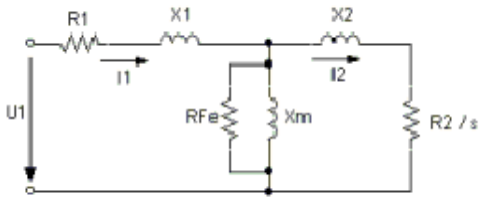


Figure 1

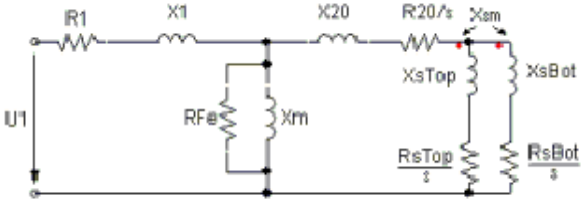


Figure 2

In the exciting branch, Xm is the magnetizing reactance, and RFe is the resistance corresponding to iron-core losses. Xm is a linearized nonlinear parameter that varies with the saturation of the main field.

After a phase voltage U1 is applied to the phase terminals, stator phase current I1 and rotor current I2, which has been referred to the stator, can be easily computed by the circuit analysis. The electromagnetic power Pm, or air-gap power, is computed by the following:

```
Pm = 3 * I2^2 * R2/s
```

The electromagnetic torque Tm is:

$$Tm = Pm/\omega$$

where w is the synchronous speed in rad/s.

The output mechanical shaft torque T2 is:

```
T2 = Tm - Tfw
```

where Tfw is the frictional and wind torque.

The output power is:

$$P2 = T2 * \omega_2$$

where $\omega_2 = \omega * (1 - s)$ and is rotor speed in rad/s.

The input power is:

$$P1 = P2 + Pfw + Pcu2 + PFe + Pcu1 + Ps$$

where Pfw, Pcu2, PFe, Pcu1, and Ps are frictional and wind loss, rotor copper loss, iron-core loss, stator copper loss, and stray loss, respectively.

The power factor is derived from:

$$PF = P1/(m * U1 * I1)$$

The efficiency is computed by:

$$eff = P2/P1 * 100\%$$

Defining a Three-Phase Induction Motor

The general procedure for defining a three-phase induction motor is as follows:

1. Insert a three-phase induction motor into an existing or new project.
2. Double-click the **Machine** entry in the project tree to define the general data, such as the number of poles and machine losses.
3. Double-click the **Machine-Stator** entry in the project tree to define the stator geometry.
4. Double-click the **Machine-Stator-Slot** entry in the project tree to define the stator slot dimensions.
5. Double-click the **Machine-Stator-Winding** entry in the project tree to define the stator windings and conductors.
6. Double-click the **Machine-Rotor** entry in the project tree to define the rotor geometry.
7. Double-click the **Machine-Rotor-Slot** entry in the project tree to define the rotor slot dimensions.
8. Double-click the **Machine-Rotor-Winding** entry in the project tree to define the rotor conductor, ventage hole dimensions, and skew.
9. Double-click the **Machine-Shaft** entry in the project tree to define the magnetism of the shaft.
10. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
11. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.

12. Choose **File>Save** to save the project.
13. Choose **RMxprt>Analyze** to analyze the design.

Note	When you place the cursor over an entry field in the data windows, a brief description of that field appears in the status bar at the bottom of the **RMxprt** window.

Once the design is analyzed, the model can be viewed in the Maxwell 2D Modeler, or it can be used to create a new Maxwell 2D project, and a new Maxwell 3D design.

Please refer to the *Three-Phase Induction Motor Problem* application note, on the technical support page of the Ansys web site, for a specific example of a three-phase induction motor problem.

Defining the General Data for a Three Phase Induction Motor

Use the **General Data Properties** window to define the basic parameters of the induction motor, such as the number of poles, and frictional loss.

To define the general data:

1. To open the **General Data Properties** window, double-click the **Machine** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the number of poles for the machine in the **Number of Poles** field. This value is the total number of poles in the stator (or the number of pole pairs multiplied by two).
3. Enter the stray loss factor in the **Stray Loss Factor** field. The stray load loss consists of the losses arising from non-uniform current distribution in the copper and additional core losses produced in the iron by distortion of the magnetic flux by the load current. The IEEE Standard provides different assumed stray load loss values for AC motors rated less than 2500 hp, as follows:
   - 1) 1-125 HP = 1.8% of rated output power
   - 2) 126-500 HP = 1.5%
   - 3) 501-2499 HP = 1.2%
4. Enter the energy loss due to friction at the given speed in the **Frictional Loss** field.
5. Enter the wind loss due to air resistance measured at the reference speed in the **Wind Loss** field.
6. Enter the given speed in the **Reference Speed** field.
7. Click **OK** to close the **Properties** window.

General Data for Three-Phase Induction Motors

To access the general data, double-click the **Machine** entry in the project tree.

The **General Data Properties** window for a three-phase induction motor contains the following fields:

Machine Type	The machine type you selected when inserting a new RMxprt design (Three Phase Induction Motor).
Number of Poles	The number of poles the machine contains.
Stray Loss Factor	The stray loss factor: the ratio of stray loss to rated output power.
Frictional Loss	The frictional energy loss (due to friction) measured at the reference speed.
Wind Loss	The wind loss (due to air resistance) measured at the reference speed.
Reference Speed	The given speed of reference.

Defining the Stator Data for a Three-Phase Induction Motor

The stator is the outer lamination stack where the three-phase windings reside. In the project tree, double-click **Machine>Stator**, **Machine-Stator-Slot**, and **Machine-Stator-Winding** to define the physical dimensions, slot data, wires, and conductors for the stator.

TTo define general stator data:

1. To open the **Stator Data Properties** window, double-click the **Machine>Stator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the **Outer Diameter** of the stator.
3. Enter the **Inner Diameter** of the stator.
4. Enter the length of the stator core in the **Length** field.
5. Enter the stacking factor for the stator core in the **Stacking Factor** field.
6. Select a **Steel Type** for the stator core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
7. Enter the **Number of Slots** in the stator.
8. Select the **Slot Type**:
   a. Click the button for the **Slot Type**.

   The **Select Slot Type** window appears.

   b. Select a slot type (available types include 1 through 4).

Note	When you place the mouse cursor over the slot type, a schematic of the selected type appears, displaying the slot dimension variables.

Optionally, check **User Defined Slot** if you wish to define the slot dimensions using the **Slot Editor**.

c. Click **OK** to close the **Select Slot Type** window and return to the **Properties** window.

9. Enter the number of sectors in the **Lamination Sectors** field.
10. Enter the thickness of the magnetic end pressboard in the **Pressboard Thickness** field. Enter **0** for a non-magnetic end pressboard.
11. Enter the skew width, measured in slot number, in the **Skew Width** field.
12. Click **OK** to close the **Properties** window.

Stator Data for Three-Phase Induction Motors

To access the general stator data, double-click the **Machine>Stator** entry in the project tree.

The **Stator Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the stator core.
Inner Diameter	The inner diameter of the stator core.
Length	The length of the stator core.
Stacking Factor	The stacking factor of the stator core.
Steel Type	The steel type of the stator core. Click the button to open the **Select Definition** window.
Number of Slots	The number of slots the stator core contains.
Slot Type	The type of slots in the stator core. Click the button to open the **Select Slot Type** window.
Lamination Sectors	The number of lamination sectors.
Pressboard Thickness	The magnetic press board thickness (0 for a non-magnetic press board).
Skew Width	The skew width measured in slot number.

Defining the Stator Slots for a Three-Phase Induction Motor

Note	If you chose **User Defined Slot** in the **Select Slot Type** window, you must define the slot dimensions using the **Slot Editor**.

To define the stator slots:

1. To open the **Stator Slot Data Properties** window, double-click the **Machine-Stator-Slot** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Optionally, to automatically design the dimensions of slots **Hs2**, **Bs1**, and **Bs2**, select the **Auto Design** check box.

3. Optionally, to design dimensions of slots **Bs1** and **Bs2** based on the stator tooth width, select the **Parallel Tooth** check box, and enter a value in the **Tooth Width** field.
4. Enter the available slot dimensions.

Hs0	Always available.
Hs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.
Bs0	Always available.
Bs1	Available only when **Auto Design** and **Parallel Tooth** are both cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Bs2	Available only when **Auto Design** and **Parallel Tooth** are both cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Rs	Rs is added when the slot type is 3 or 4.

5. Click **OK** to close the **Properties** window.

Stator Slot Data for Three-Phase Induction Motors

To access the stator slot data, double-click the **Machine-Stator-Slot** entry in the project tree.

The **Stator Slot Data Properties** window contains the following fields:

Auto Design	Select or clear this to enable or disable auto-design of slots **Hs2**, **Bs1**, and **Bs2**. When this check box is selected, only two other fields appear in the window: **Hs0** and **Bs0**.
Parallel Tooth	Select this to design **Bs1** and **Bs2** based on the tooth width. When this check box is selected, the **Bs1** and **Bs2** fields are removed, and the **Tooth Width** field is added.
Tooth Width	The tooth width for the parallel tooth, on which **Bs1** and **Bs2** are designed.
Hs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Hs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Rs	A slot dimension. (see the diagram shown in the modeling window when **Machine-**

	Stator-Slot is selected). Rs is added when the slot type is 3 or 4.

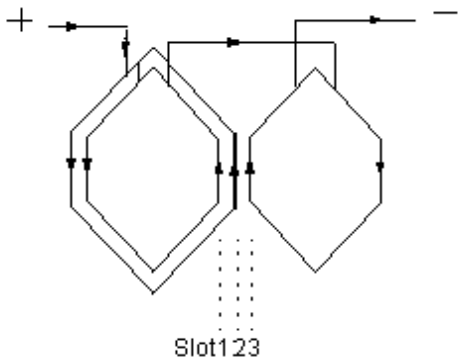
Defining the Stator Windings for a Three-Phase Induction Motor

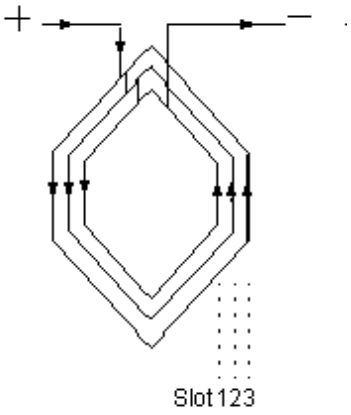
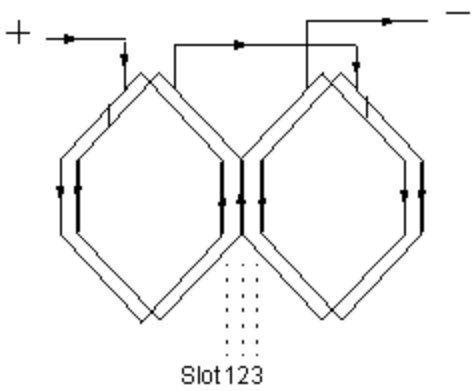
Define the wires, conductors, insulation, and windings of the stator.

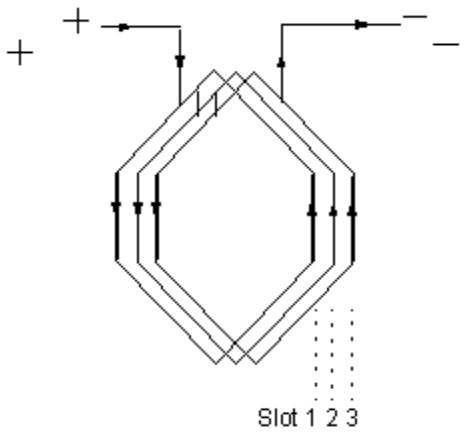
To define the wires and windings:

1. To open the **Stator Slot Winding Properties** window, double-click the **Machine-Stator-Winding** entry in the project tree on the desktop. You can also enter values in the **Properties** section of the desktop without opening a separate window.
2. Click the **Winding** tab.
3. Enter the number of layers in the stator winding in the **Winding Layers** field.
4. Select a **Winding Type**:
   a. Click the button for **Winding Type**.
   The **Winding Type** window appears.
   b. Select from one of the following three types of winding:
      - **Whole Coiled**
      - **Half Coiled**
      - **Editor**

When you place the mouse cursor over a winding button, an outline of the selected winding appears. The following table describes the six types of windings that are possible (three for one-layer and three for two-layer):

Type	Description
One layer winding Editor	A user-defined one-layer winding arrangement. You need to set up the winding arrangement for each slot.
Whole Coiled	A one-layer whole-coiled winding: 
Half Coiled	A one-layer concentric half-coiled winding:

	
Two Layer Winding Editor	A user-defined two-layer winding arrangement. When you select for winding layers the you can specify a different winding arrangement for each slot in the Winding Editor.
Whole Coiled	A two-layer whole coiled winding:  The phase belt for this winding configuration is equal to 360/2*m*, where *m* is the phase number.
Half Coiled	A two-layer half-coiled winding:



There is only one coil per phase per pair of poles.

Note	For a two-layer winding, if you check **Constant Pitch** in the **Winding Editor**, only the top layer needs to be defined; the bottom layer will be determined according to the coil pitch. Once you have clicked a button to select a winding, click **OK** to close the **Winding Type** window and return to the **Properties** window.

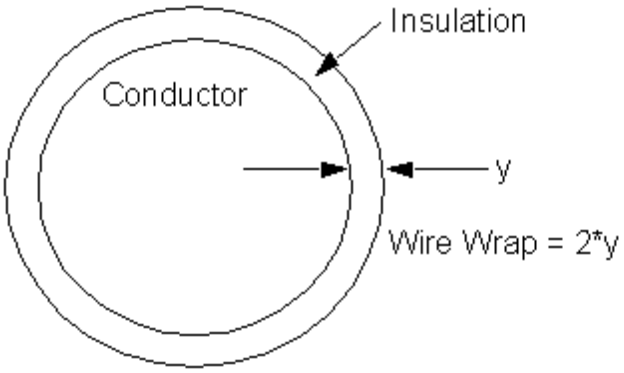
a. Select or enter the number of parallel branches in one phase of the winding in the **Parallel Branches** field.

b. Enter the total number of conductors in each stator slot in the **Conductors per Slot** field. This value is the number of turns per coil multiplied by the number of layers.

c. Enter the coil pitch, measured in number of slots, in the **Coil Pitch** field. The coil pitch is the number of slots separating one winding. For example, if a coil starts in slot 1 and ends in slot 6, it has a coil pitch of 5.

d. Enter the number of wires per conductor in the **Number of Strands** field. Enter 0 to have RMxprt auto-design this value.

e. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



f. Select the **Wire Size**: Click the button for **Wire Size**. The **Wire Size** window appears.
g. Select a value from the **Wire Diameter** pull-down list.
h. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

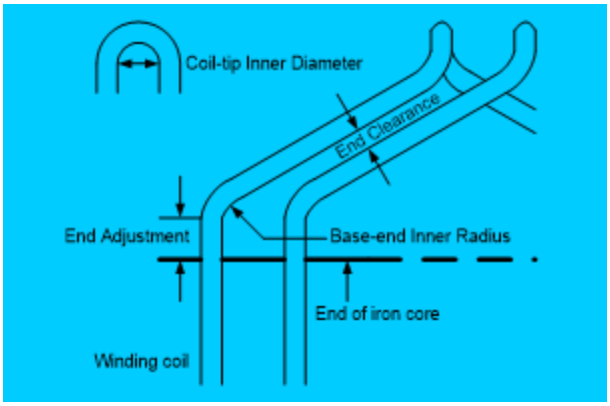
<number>	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.
USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.
AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.
MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21 mm and 2 with a diameter of 0.13 mm.

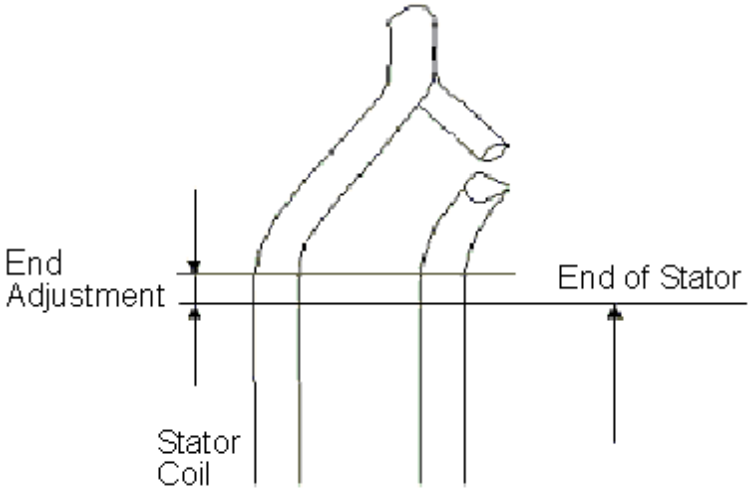
The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>General Options>Machines**command.

When you are done setting the wire size, click **OK** to close the **Wire Size** window and return to the **Properties** window.

a. Click the **End/Insulation** tab.
b. Select or clear the **Input Half-turn Length** check box.
c. Do one of the following:
   - If you selected **Input Half-turn Length**, then enter the half-turn length of the armature winding in the Half Turn Length field.
   - If you cleared **Input Half-turn Length**, then enter the end length adjustment of the stator coils in the **End Adjustment** field. The end adjustment is the distance one end of the

conductor extends vertically beyond the end of the stator.





5. Enter the inner radius of the base corner in the **Base Inner Radius** field.
6. Enter the inner diameter of the coil tip in the **Tip Inner Diameter** field.
7. Enter the distance between two stator coils in the **End Clearance** field.
8. Enter the thickness of the slot liner insulation in the **Slot Liner** field.
9. Enter the thickness of the wedge insulation in the **Wedge Thickness** field.
10. Enter the thickness of the insulation layer in the **Layer Insulation** field.
11. Enter the limited slot fill factor for the wire design in the **Limited Fill Factor** field. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.
12. Click **OK** to close the **Properties** window.

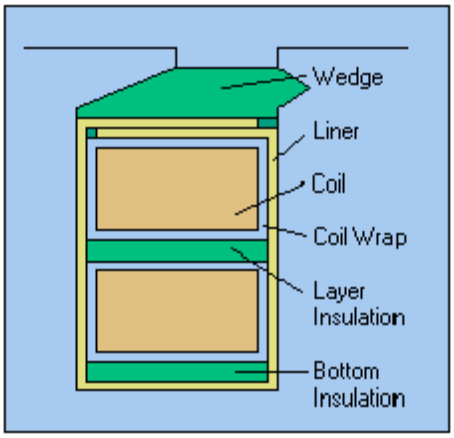
Stator Winding Data for Three-Phase Induction Motors

To access the stator winding data, double-click the **Machine-Stator-Winding** entry in the project tree.

The **Stator Winding Data Properties** window contains the following fields:

Winding tab	**Winding Layers**	The number of winding layers.
	Winding Type	The type of stator winding. Click the button to open the **Winding Type** window and choose from **Whole Coiled**, **Half Coiled**, and **Editor**.
	Parallel Branches	The number of parallel branches in the stator winding.
	Conductors per Slot	The number of conductors per stator slot (0 for auto-design).
	Coil Pitch	The coil pitch measured in number of slots.
	Number of Strands	The number of wires per conductor (0 for auto-design).
	Wire Wrap	The thickness of the double-sided wire wrap (0 to automatically obtain this value from the wire library).
	Wire Size	The diameter of the wire (0 for auto-design). Click the button to open the **Wire Size** window where you can specify units, wire type, diameter, and gauge.
End/Insulation tab	**Input Half-turn Length**	Select or clear this check box to specify whether or not you want to enter the half-turn length. When this check box is selected, the **Half Turn Length** field appears the next time you open the **Properties** window. When this check box is selected, the **End Adjustment** field appears instead.
	Half Turn Length	The half-turn length of the armature winding.
	End Adjustment	The end length adjustment of the stator coils, which is the distance one end of the conductor extends vertically beyond the end of the stator.
	Base Inner Radius	The inner radius of the base corner.

	Tip Inner Diameter	The inner diameter of the coil tip.
	End Clearance	The end clearance between two adjacent coils.
	Slot Liner	The thickness of the slot liner insulation.
	Wedge Thickness	The thickness of the wedge insulation.
	Layer Insulation	The thickness of the insulation layer.
	Limited Fill Factor	The limited slot fill factor for the wire design. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.



Winding Editor for a Three-Phase Induction Motor

For a three-phase induction motor, you may want to specify a different number of conductors for each stator slot. The **Winding Editor** makes this possible by enabling you to specify the number of turns for each coil. To enable the **Winding Editor**, you must have set the **Winding Property** for the **Winding Type** to **Editor**.

To specify the number of turns for each coil:

1. Click **Machine>Winding>Edit Layout**.
   The **Winding Editor** window appears.
2. In the table in the upper left, set which phase you want for each coil and which slot is the “in” and “out” slot for the current in each coil.
3. If you are working on a quarter or half model, you may want to specify a multiplier by clicking the **Periodic Multiplier** check box and specifying a value.

4. Select or deselect the **Constant Turns** or **Constant Pitch** check boxes, depending on whether you want to be able to change these setting in the table above. When these options are selected, you cannot change the turns or pitch.
5. When you are satisfied with the coil settings, click **OK** to close the **Winding Editor** window.

Defining Different Size Wires for a Three-Phase Induction Motor

Use the **Gauge** option if you have a conductor that is made up different size wires.

To define different size wires:

1. In the **Wire Size** window, select **MIXED** from the **Gauge** pull-down menu.
2. Select either **Round** or **Rectangular** as the **Wire Type**.
3. Enter the appropriate wire data in the table:
   - For a round wire:
     - Enter the **Diameter** in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this diameter.
   - For a rectangular wire:
     - Enter the **Width** of the wire in the table.
     - Enter the **Thickness** of the wire in the table.
     - Enter the **Fillet** value in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this data.
4. Choose **Add** to add the new wire data.
5. Repeat steps 3 and 4 for each size wire you want to add.
6. When you are finished defining the wires, click **OK** to close the **Wire Size** window.

Note	For example, if one conductor is made up of 5 wires, and 3 of those wires have a diameter of 0.21mm, and the other 2 have a diameter of 0.13mm, then the mixed wire size table will have two lines. The first line will list Diameter = 0.21 and Number = 3. The second line will list Diameter = 0.13 and Number = 2. An equivalent wire diameter is displayed as Wire Size value in the Winding tab in the Properties window.

Stator Vent Data for Three-Phase Induction Motors

By option, you can add a vent to a three-phase induction motor. To add a vent to stator in a three-phase induction motor.

1. Select the stator icon in the project tree.
2. Right-click to display the pop-up menu and select **Insert Vent**.

   The vent icon appears in the project tree under the stator.

To remove a vent to stator in a three-phase induction motor.

1. Select the stator icon in the project tree.
2. Right-click to display the pop-up menu and select **Remove Vent**.
   The vent icon disappears in the project tree under the stator.

The Vent data for the stator includes the following fields.

Vent Ducts	Number of radial vent ducts
Duct Width	Width of radial vent ducts
Magnetic Spacer Width	Width of magnetic spacer which holds vent ducts. 0 for non magnetic spacer
Duct Pitch	Center-to-Center distance between two adjacent Vent ducts

Defining the Rotor Data for a Three-Phase Induction Motor

The rotor consists of copper bars in which current is induced by the magnetic fields produced by the stator windings. In the project tree, double-click **Machine-Rotor**, **Machine-Rotor-Slot**, and **Machine-Rotor-Winding** to define the rotor slots and vents.

To define general stator data:

1. To open the **Rotor Data Properties** window, double-click the **Machine>Rotor** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the stacking factor for the rotor core in the **Stacking Factor** field.
3. Enter the **Number of Slots** in the rotor.
4. Select the **Slot Type**:
   a. Click the button for the **Slot Type**.
      The **Select Slot Type** window appears.
   b. Select a slot type (available types include 1 through 4).

Note	When you place the mouse cursor over the slot type, a schematic of the selected type appears, displaying the slot dimension variables.

   c. Click **OK** to close the **Select Slot Type** window and return to the **Properties** window.
5. Enter the outer diameter of the rotor in the **Outer Diameter** field.
6. Enter the inner diameter of the rotor in the **Inner Diameter** field.
7. Enter the length of the rotor core in the **Length** field.
8. Select a **Steel Type** for the rotor core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.

9. Enter the **Skew Width**, measured in rotor slot pitch. This value defines by how much the rotor bars are skewed.
10. Optionally, select **Cast Rotor** to allow the conductor to fill all the space available in the slot. Otherwise, RMxprt assumes the slot wedge that fixes the bars is filled with insulator material in a 2D/3D geometry model.
11. Optionally, select **Half Slot** to draw only half of the rotor slots.
12. Optionally, select **Double Cage** to specify the winding as a double-squirrel-cage winding.

    If you select Double Cage, another line appears in the properties to let you specify the Bottom Slot type.

    a. Click on the Custom button on the Double Cage row. This displays the Select Slot Type window.
    The **Select Slot Type** window appears.
    b. Select a slot type (available types include 1 through 4).

Note	When you place the mouse cursor over the slot type, a schematic of the selected type appears, displaying the slot dimension variables.

13. Click **OK** to close the **Properties** window.

Rotor Data for Three-Phase Induction Motors

To access the general rotor data, double-click the **Machine>Rotor** entry in the project tree.

The **Rotor Data Properties** window contains the following fields:

Stacking Factor	The stacking factor of the rotor core.
Number of Slots	The number of slots the rotor core contains.
Slot Type	The type of slots in the rotor core. Click the button to open the **Select Slot Type** window.
Outer Diameter	The outer diameter of the rotor core.
Inner Diameter	The inner diameter of the rotor core.
Length	The length of the rotor core.
Steel Type	The steel type of the rotor core. Click the button to open the **Select Definition** window.
Skew Width	The skew width measured in slot number.
Cast Rotor	Select or clear this to specify whether the rotor squirrel-cage winding is cast or not.
Half Slot	Select this to specify a half-shaped unsymmetrical slot.
Double Cage	Select this to specify the winding as double-squirrel-cage.

Defining the Rotor Slots for a Three-Phase Induction Motor

To define the type and dimensions of the rotor's slots:

1. To open the **Rotor Data Slot Properties** window, double-click the **Machine-Rotor-Slot** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the slot dimensions in the following fields: **Hs0**, **Hs01**, **Hs2**, **Bs0**, **Bs1**, **Bs2**, **Rs**.
3. Click **OK** to close the **Properties** window.

Rotor Slot Data for Three-Phase Induction Motors

To access the rotor slot data, double-click the **Machine-Rotor-Slot** entry in the project tree.

The **Rotor Slot Data Properties** window contains the following fields:

Hs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Hs01	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Hs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Bs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Bs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Bs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Rs	A slot dimension. (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected). Rs is added when the slot type is 3 or 4.

Defining the Rotor Winding for a Three-Phase Induction Motor

To define the rotor winding data:

1. To open the **Rotor Data Slot Properties** window, double-click the **Machine-Rotor-Winding** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select a **Bar Conductor Type** for the rotor winding bar:
   a. Click the button for **Bar Conductor Type**.
   The **Select Definition** window appears.
   b. Select a conductor type from the list, or define a new conductor type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.

3. Enter the length of the gap between the end ring and the iron core in the **End Length** field. This field specifies the value for only one end of the gap, not both.
4. Enter the end ring dimension in the axial direction in the **End-Ring Width** field. The end ring connects the bars of the rotor to one another.
5. Enter the end ring dimension in the radius direction in the **End-Ring Height** field. The end ring's height covers at least the cross-section of the rotor conductor.
6. Select an **End Ring Conductor Type** for the rotor winding end ring:
   a. Click the button for **End Ring Conductor Type**.
      The **Select Definition** window appears.
   b. Select a conductor type from the list, or define a new conductor type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
7. Click **OK** to close the **Properties** window.

Rotor Winding for Three-Phase Induction Motors

To access the rotor winding data, double-click the **Machine-Rotor-Winding** entry in the project tree.

The **Rotor Winding Data Properties** window contains the following fields:

Bar Conductor Type	The type of bar conductor used in the winding. Click the button to open the **Select Definition** window.
End Length	The length of the single-side end of the extended bar.
End Ring Width	The width of one side of the end rings in the axial direction. The end ring connects the bars of the rotor to one another.
End Ring Height	The height of the end rings in the radian direction. The end ring connects the bars of the rotor to one another.
End Ring Conductor Type	The type of end ring conductor used in the winding. Click the button to open the **Select Definition** window.

Rotor Vent Data for Three-Phase Induction Motors

By option, you can add a vent to a rotor in a three-phase induction motor.

To add a vent to rotor:

1. Select the rotor icon in the project tree.
2. Right-click to display the pop-up menu and select **Insert Vent**.
   The vent icon appears in the project tree under the rotor.

To remove a vent to stator in a three-phase induction motor.

1. Select the rotor icon in the project tree.
2. Right-click to display the pop-up menu and select **Remove Vent**.
   The vent icon disappears in the project tree under the stator.

The Vent data for the rotor includes the following fields.

Vent Ducts	Number of radial vent ducts
Duct Width	Width of radial vent ducts
Magnetic Spacer Width	Width of magnetic spacer which holds vent ducts. 0 for non magnetic spacer
Duct Pitch	Center to center distance between vent ducts
Holes per row	Number of axial vent holes per row.
Inner hole diameter	Diameter of vent holes in inner row.
Outer hole diameter	Diameter of vent holes in outer row.
Inner hole location	Center to center diameter of inner hole vents
Outer hole location	Center to center diameter of outer hole vents.

Defining the Shaft Data for a Three-Phase Induction Motor

To define the shaft:

1. To open the **Shaft Data Properties** window, double-click the **Machine>Shaft** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select or clear the **Magnetic Shaft** check box to specify whether or not the shaft is to be made of magnetic material.
3. Click **OK** to close the **Properties** window.

Shaft Data for Three-Phase Induction Motors

To access the shaft data, double-click the **Machine>Shaft** entry in the project tree.

The **Shaft Data Properties** window contains the following fields:

Magnetic Shaft	Select or clear this check box to indicate whether or not the shaft is made of magnetic material.

Setting Up Analysis Parameters for a Three-Phase Induction Motor

To define the solution data:

1. To open the Solution Setup window, right-click **Analysis** in the project tree, and click **Add Solution Setup**.
2. Click the **General** tab. The **Operation Type** is automatically set to **Motor** for this machine type.

3. Select the **Load Type** used in the motor from the following options:

Const Speed	The speed remains constant in the motor.
Const Power	The output power remains constant in the motor.
Const Torque	The torque remains constant regardless of the speed. In this case, $T_{load} = T_{rated}$, given by the output power divided by the given rated speed.
Linear Torque	The torque increases linearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})$ where T_{rated} is given by the output power divided by the given rated speed.
Fan Load	The load varies nonlinearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})^2$ where T_{rated} is given by the output power divided by the given rated speed.

4. Enter the output power developed at the shaft of the motor in the **Rated Output Power** field.
5. Enter the RMS line-to-line voltage in the **Rated Voltage** field.
6. Enter the desired rotor speed in the **Rated Speed** field. When input Rated Speed is less than the Synchronous Speed, the machine operates as a motor. When input Rated Speed is greater than the Synchronous Speed, the machine operates as a generator. For an induction generator, the rated performance will be calculated at the input Rated Speed with the three-phase windings connecting to infite bus.
7. Enter the temperature at which the system functions in the **Operating Temperature** field.
8. Click the **Three-Phase Induction Motor** tab.
9. Enter the electrical line frequency in the **Frequency** field, and select the units.
10. Select the **Winding Connection** from the following options:
    - **Wye** (Y)
    - **Delta**
11. Click **OK** to close the **Solution Setup** window.

Related Topics

Solution Data for Three-Phase Induction Motors

Solution Data for Three-Phase Induction Motors

To access the solution data, right-click **Analysis** in the project tree, and click **Add Solution Setup**.

The **Solution Setup** window contains the following fields:

Operation Type	*On the* **General** *tab.* The operation type is automatically set to **Motor** for this machine type.
Load Type	*On the* **General** *tab.* Select from **Const Speed**, **Const Power**, **Const Torque**, **Linear Torque**, and **Fan Load**. The default is **Const Power**.

Rated Output Power	*On the* **General** *tab.* Type a value for the rated output voltage, and select the units.
Rated Voltage	*On the* **General** *tab.* Type a value for the rated voltage, and select the units.
Rated Speed	*On the* **General** *tab.* Type a value for the rated speed, and select the units.
Operating Temperature	*On the* **General** *tab.* Type a value for the operating temperature, and select the units.
Frequency	*On the* **Three-Phase Induction Motor** *tab.* Type a value for the frequency, and select the units.
Winding Connection	*On the* **Three-Phase Induction Motor** *tab.* Select from **Wye** or **Delta**.

Related Topics

Setting Up Analysis Parameters for a Three-Phase Induction Motor

Single-Phase Induction Motors

After you have selected **Single-Phase Induction Motors** as your model type, you must define the following:

- General data, such as the voltage, speed, and materials used in the motor.
- Stator data, such as the slot types, diameter, and wire dimensions.
- Rotor data, such as the slot dimensions, skew width, and ventage holes.
- Solution data, such as rated output voltage and frequency.

By option, you can:

- Add a machine housing.
- Add a vent or remove a vent from the rotor.

Analysis Approach for Single-Phase Induction Motors

The construction of a single-phase induction motor is structurally similar to the poly-phase squirrel-cage induction motors. The primary difference is that the stator windings, which consist of a main winding and an auxiliary winding, have axes of these that are displaced 90 electrical degrees in space. To produce a starting torque, the currents in the two windings must be out of phase. Usually a capacitor is connected in series with the auxiliary winding so that the auxiliary winding current is forced to lead the main winding current by about 90 electrical degrees. Two parallel capacitors can also be used: one for starting, and one for running, so that both a starting and running performance are obtained.

An algorithm called symmetric component method is applied to analyze single-phase induction motors (IndM1). Both voltages and currents of the main-phase and auxiliary-phase windings are decomposed to positive- and negative-sequence components. The equivalent circuits for main-phase positive-sequence components, auxiliary-phase positive-sequence components, main-

phase negative-sequence components, and auxiliary-phase negative-sequence components are shown in (a), (b), (c), and (d) of Figure 3, respectively.

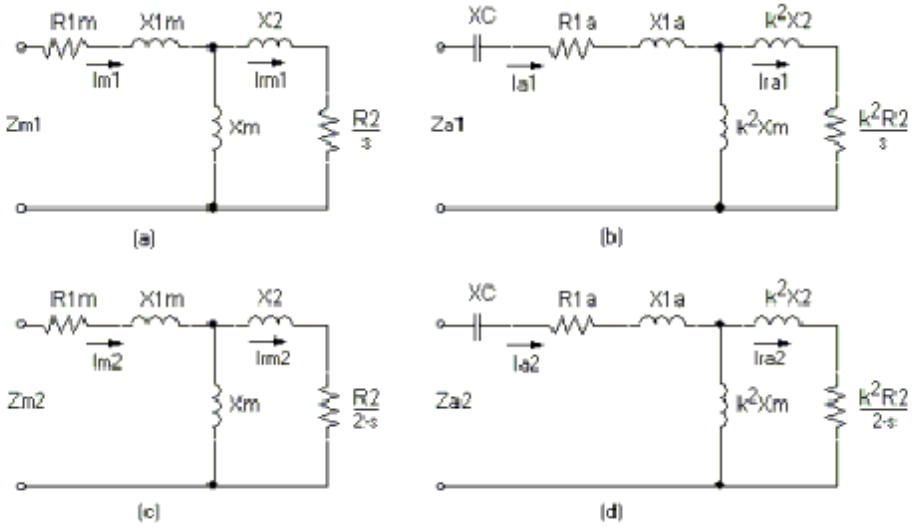


Figure 3

In the figures, R1m, X1m, R1a, X1a, R2, X2, and Xm are main-phase stator resistance, main-phase stator leakage reactance, auxiliary-phase stator resistance, auxiliary-phase stator leakage reactance, rotor resistance, rotor leakage reactance, and magnetizing reactance, respectively. XC is the reactance of the capacitor connected in series with the auxiliary winding, and the coefficient k is the ratio of effective turns of the auxiliary winding to that of the main winding. R2, X2, and Xm have been referred to the main winding. The equivalent impedance of the four circuits is Z_{m1}, Z_{a1}, Z_{m2}, and Z_{a2}, as shown in the figures.

According to the symmetric component method, the positive and negative components of auxiliary-phase currents can be expressed in the form of a phasor as the following:

$$\mathbf{I}_{a1} = (j / k)\mathbf{I}_{m1}$$

$$\mathbf{I}_{a2} = ((j / k)\mathbf{I}_{m2}$$

Because the main winding and the auxiliary winding have the same applied terminal voltage U1, the voltage equations for both windings become the following:

$$\mathbf{U}_1 = \mathbf{U}_{m1} + \mathbf{U}_{m2} = \mathbf{I}_{m1}\mathbf{Z}_{m1} + \mathbf{I}_{m2}\mathbf{Z}_{m2}$$

$$\mathbf{U}_1 = \mathbf{U}_{a1} + \mathbf{U}_{a2} = \mathbf{I}_{a1}Z_{a1} + \mathbf{I}_{a2}Z_{a2} = (j / k)(\mathbf{I}_{m1}Z_{a1} - \mathbf{I}_{m2}Z_{a2})$$

The positive and negative components of main-phase current are calculated by the following:

$$\mathbf{I}_{m1} = \mathbf{U}_1(Z_{a2} - jkZ_{m2}) / (Z_{m1}Z_{a2} + Z_{m2}Z_{a1})$$

$$\mathbf{I}_{m2} = \mathbf{U}_1(Z_{a1} + jkZ_{m1}) / (Z_{m1}Z_{a2} + Z_{m2}Z_{a1})$$

The total input current is:

$$\mathbf{I}_1 = \mathbf{I}_m + \mathbf{I}_a = (\mathbf{I}_{m1} + \mathbf{I}_{m2}) + (\mathbf{I}_{a1} + \mathbf{I}_{a2})$$

Based on these two components of main-phase current, all current components shown in Figure 3 can be obtained by simple computation.

Then the total input current is:

$$\mathbf{I}_1 = \mathbf{I}_m + \mathbf{I}_a = (\mathbf{I}_{m1} + \mathbf{I}_{m2}) + (\mathbf{I}_{a1} + \mathbf{I}_{a2})$$

The positive- and negative-sequence air-gap power can be computed in the following way:

```
Pm1 = 2 * Irm1^2 * R2 / s
Pm2 = 2 * Irm2^2 * R2 / (2 - s)
```

The total air-gap power is:

```
Pm = Pm1 - Pm2
```

Tm, T2, P2, P1, and eff are computed in the same way as for three-phase induction motors.

The power factor is derived from:

```
PF = P1 / (U1 * I1)
```

Defining a Single-Phase Induction Motor

The general procedure for structurally defining a single-phase induction motor is as follows:

1. Insert a single-phase induction motor design in an existing or newly created project.
2. Double-click the **Machine** entry in the project tree to define the general data.
3. Double-click the **Machine-Stator** entry in the project tree to define the stator geometry.
4. Double-click the **Machine-Stator-Slot** entry in the project tree to define the stator slot dimensions.
5. Double-click the **Machine-Stator-Winding** entry in the project tree to define the stator windings and conductors.
6. Double-click the **Machine-Rotor** entry in the project tree to define the rotor geometry.
7. Double-click the **Machine-Rotor-Slot** entry in the project tree to define the rotor slot dimensions.
8. Double-click the **Machine-Rotor-Winding** entry in the project tree to define the rotor conductor, ventage hole dimensions, and skew.
9. Double-click the **Machine-Shaft** entry in the project tree to define the magnetism of the shaft.
10. Optionally, for Inner Rotor machines only, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
11. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
12. Choose **File>Save** to save the project.
13. Choose **RMxprt>Analyze** to analyze the design.

Note	When you place the cursor over an entry field in the data windows, a brief description of that field appears in the status bar at the bottom of the **RMxprt** window.

Once the design is analyzed, the model can be imported into the Maxwell 2D Modeler, or can be used to create a new Maxwell 2D project, and a new Maxwell 3D design.

Please refer to the *A Capacitor-Run Single-Phase Induction Motor Problem* application note, on the technical support page of the Ansys web site, for a specific example of a single-phase induction motor problem.

Defining the General Data for a Single-Phase Induction Motor

Use the **General Data Properties** window to define the basic parameters of the induction motor, such as the number of poles, frictional loss, and operation mode.

To define the general data:

1. To open the **General Data Properties** window, double-click the **Machine** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the number of poles for the machine in the **Number of Poles** field. This value is the total number of poles in the stator (or the number of pole pairs multiplied by two).
3. Select one of the following for the **Rotor Position**:
   - **Inner Rotor**
   - **Outer Rotor**
4. Enter the energy loss due to friction at the given speed in the **Frictional Loss** field.
5. Enter the wind loss due to air resistance measured at the reference speed in the **Wind Loss** field.
6. Enter the given speed in the **Reference Speed** field.
7. Select the **Operation Mode**:
   a. Click the button.
   The **Select Operation Mode** window appears.
   b. Select from one of the following:

C-Run	Capacitance-run mode. The capacitor is in series with the auxiliary winding. In the Capacitor run mode, the capacitor will be designed (if the auto-design mode is selected) to minimize the backward magnetomotive force.
C-Start	Capacitance-start mode. The auxiliary winding is in series with the capacitor and is disconnected when the rotor reaches the switching speed.
C-R&S	Capacitance-run and start mode. Two capacitors are in series with the auxiliary winding; one for starting, one for running.
R-Start	Resistor-start mode. The auxiliary winding is disconnected when the rotor reaches the switching speed.

   c. Click **OK** to close the **Select Operation Mode** window and return to the **Properties** window.

8. Enter values in the following capacitance, resistance, and switching speed fields:

Run Capacitance	Available for C-Run, C-R&S
Run Resistance	Available for C-Run, C-R&S
Start Capacitance	Available for C-Start, C-R&S
Start Resistance	Available for C-Start, C-R&S
Switching Speed	Available for C-Start, C-R&S, R-Start

9. If the start winding needs to be optimized, select the **Objective Type** from the following three options:
   - **(Tst/Ist)max**. Accept the defaults. This is the ratio of the maximum starting torque to the starting current ratio.
   - **(Tst)max**. Enter the given start current ratio. This is the maximum starting torque
   - **(Ist)min (minimum starting current). Enter the given start torque ratio.**

Note	The start-winding optimization goal is disabled for the **C-Run** operation mode. In capacitor-run mode, the capacitor is designed to minimize the backward magnetomotive force. For other modes, if the auto-design function is active, the capacitor and the resistance are designed according to the start goal, selected from the following: • The maximum value of (Starting Torque/Starting Current). • The maximum starting torque. • The minimum starting current.

10. Click **OK** to close the **Properties** window.

General Data for Single-Phase Induction Motors

To access the general data, double-click the **Machine** entry in the project tree.

The **General Data Properties** window for a three-phase induction motor contains the following fields:

Machine Type	The machine type you selected when inserting a new RMxprt design (Single Phase Induction Motor).
Number of Poles	The number of poles the machine contains.
Rotor Position	Select whether the rotor is an **Inner Rotor** or **Outer Rotor**.
Frictional Loss	The frictional energy loss (due to friction) measured at the reference speed.
Wind Loss	The wind loss (due to air resistance) measured at the reference speed.
Reference Speed	The given speed of reference.

Operation Mode	Click the button to select from the following four modes: **C-Run**, **C-Start**, **C-R&S**, and **R-Start**.
Run Capacitance	The capacitance of the run capacitor. Available for C-Run and C-R&S operation modes.
Run Resistance	The resistance of the run capacitor. Available for C-Run and C-R&S operation modes. See **Note** below.
Start Capacitance	The resistance of the start capacitor. Available for C-Start and C-R&S operation modes.
Start Resistance	The resistance of the start capacitor. Available for C-Start and C-R&S operation modes.
Switching Speed	The switching speed of the capacitor or resistor. Available for C-Start, C-R&S, and R-Start operation modes.
Objective Type	If the start winding needs to be optimized, select from the following three objective types: **(Tst/Ist)max**, **(Tst)max**, or **(Ist)min**. • For **(Tst/Ist) max**, accept the defaults. This is the ratio of the maximum starting torque to the starting current ratio. • For **(Tst) max**, enter the **Given Start Current Ratio**. This is the maximum starting torque. • For **(Ist) min** (minimum starting current), enter the **Given Start Torque Ratio**. The start-winding optimization goal is disabled for the **C-Run** operation mode.

Note	When exporting the RMxprt model to Maxwell: • If the value of the **Run Resistance** is **zero** in RMxprt, the value of the **Run Resistance** will be autocomputed in Maxwell to a value of 1% of the capacitor reactance. • To neglect the **Run Resistance** in Maxwell, set the value to a small non-zero number in RMxprt.

Defining the Stator Data for a Single-Phase Induction Motor

The stator is the outer lamination stack where the three-phase windings reside. In the project tree, double-click **Machine-Stator**, **Machine-Stator-Slot**, and **Machine-Stator-Winding** to define the physical dimensions, slot data, wires, and conductors for the stator.

To define general stator data:

1. To open the **Stator Data Properties** window, double-click the **Machine>Stator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the **Outer Diameter** of the stator.
3. Enter the **Inner Diameter** of the stator.
4. Enter the length of the stator core in the **Length** field.

5. Enter the stacking factor for the stator core in the **Stacking Factor** field.
6. Select a **Steel Type** for the stator core:
   a. Click the button for **Steel Type**.
   The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
7. Enter the **Number of Slots** in the stator.
8. Select the **Slot Type**:
   a. Click the button for the **Slot Type**.
   The **Select Slot Type** window appears.
   b. Select a slot type (available types include 1 through 4).

Note	When you place the mouse cursor over the slot type, a schematic of the selected type appears, displaying the slot dimension variables.

   c. Click **OK** to close the **Select Slot Type** window and return to the **Properties** window.
9. Enter the overall width of the stator outer profile in the **Overall Width** field.
10. Click **OK** to close the **Properties** window.

Stator Data for Single-Phase Induction Motors

To access the general stator data, double-click the **Machine>Stator** entry in the project tree.

The **Stator Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the stator core.
Inner Diameter	The inner diameter of the stator core.
Length	The length of the stator core.
Stacking Factor	The stacking factor of the stator core.
Steel Type	The steel type of the stator core. Click the button to open the **Select Definition** window.
Number of Slots	The number of slots the stator core contains.
Slot Type	The type of slots in the stator core. Click the button to open the **Select Slot Type** window.
Overall Width	The overall width of the stator outer profile.

Defining the Stator Slots for a Single-Phase Induction Motor

Use the **Stator1** window to define the physical dimensions of the stator slots.

To define the stator slots:

1. To open the **Stator Slot Data Properties** window, double-click the **Machine-Stator-Slot** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Optionally, to automatically design the dimensions of slots **Hs2**, **Bs1**, and **Bs2**, select the **Auto Design** check box.
3. Optionally, to design dimensions of slots **Bs1** and **Bs2** based on the stator tooth width, select the **Parallel Tooth** check box, and enter a value in the **Tooth Width** field.
4. Enter the available slot dimensions.

Hs0	Always available.
Hs1	Always available.
Hs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.
Bs0	Always available.
Bs1	Available only when **Auto Design** and **Parallel Tooth** are both cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Bs2	Available only when **Auto Design** and **Parallel Tooth** are both cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Rs	Rs is added when the slot type is 3 or 4.

5. Click **OK** to close the **Properties** window.

	Parallel Branches	The number of parallel branches in the series winding.
	Number of Strands	The number of wires per conductor in the series winding (0 for auto-design).
	Wire Wrap	The thickness of the double-sided wire wrap (0 for auto-pickup from the wire library).
	Wire Size	The wire diameter (0 for auto-design).

Stator Slot Data for Single-Phase Induction Motors

To access the stator slot data, double-click the **Machine-Stator-Slot** entry in the project tree.

The **Stator Slot Data Properties** window contains the following fields:

Auto Design	Select or clear this to enable or disable auto-design of slots **Hs2**, **Bs1**, and **Bs2**. When this check box is selected, only two other fields appear in the window: **Hs0** and **Bs0**.

Parallel Tooth	Select this to design **Bs1** and **Bs2** based on the tooth width. When this check box is selected, the **Bs1** and **Bs2** fields are removed, and the **Tooth Width** field is added.
Tooth Width	The tooth width for the parallel tooth, on which **Bs1** and **Bs2** are designed.
Hs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Hs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Hs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Rs	A slot dimension. (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected). Rs is added when the slot type is 3 or 4.

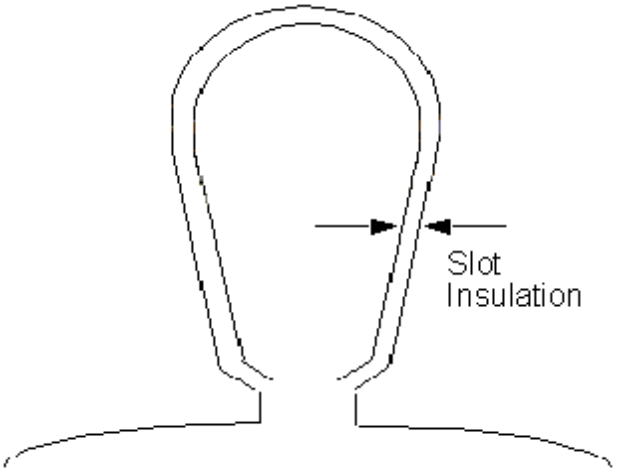
Defining the Stator Windings for a Single-Phase Induction Motor

Define the wires, conductors, insulation, and windings of the stator.

To define the wires and windings:

1. To open the **Stator Winding Properties** window, double-click the **Machine-Stator-Winding** entry in the project tree on the desktop. You can also enter values in the **Properties** section of the desktop without opening a separate window.
2. Click the **Winding** tab.

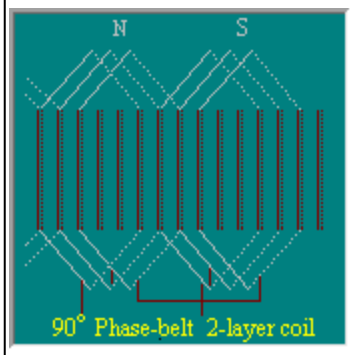
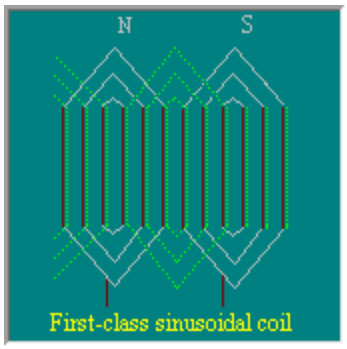
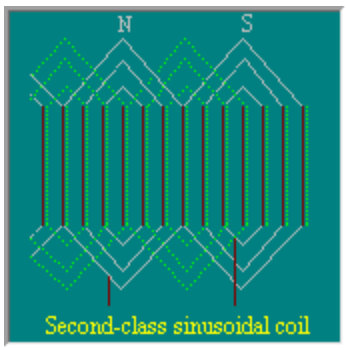
3. Enter the thickness of the slot liner in the **Slot Liner** field.



4. Enter the thickness of the wedge insulation in the **Wedge Thickness** field.
5. Enter the limited slot fill factor for the wire design in the **Limited Fill Factor** field. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.
6. Select or clear the **Include Series Winding** check box. This option sets whether or not to include the series winding in the speed adjustment. When this option is selected, a third tab, **Series (C)**, appears in the **Properties** window.
7. Enter the number of layers in the **Winding Layers** field.
8. Enter the number of slots in the **Coil Pitch** field.
9. Select a **Winding Type**:
   a. Click the button for **Winding Type**.
   The **Winding Type** window appears.
   b. Select from one of the following types of winding:
      - **Lap**
      - **Sin_1**
      - **Sin_2**
      - **Editor**

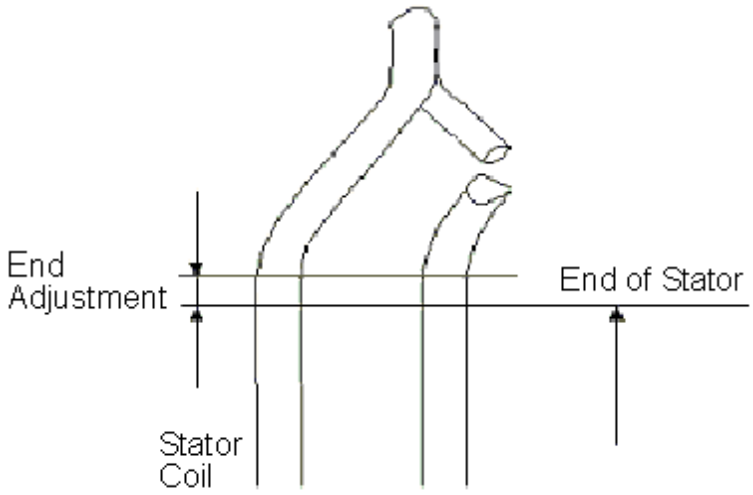
   When you place the mouse cursor over a winding button, an outline of the selected winding appears. The following table describes the four types of windings that are possible:

Editor	A user-defined one or two-layer winding arrangement, determined by your **Winding Layers** selection. Enables the Winding Editor, where you can specify a different winding arrangement for each slot.
Lap	90 deg phase belt 2-layer coil for both single and double layer

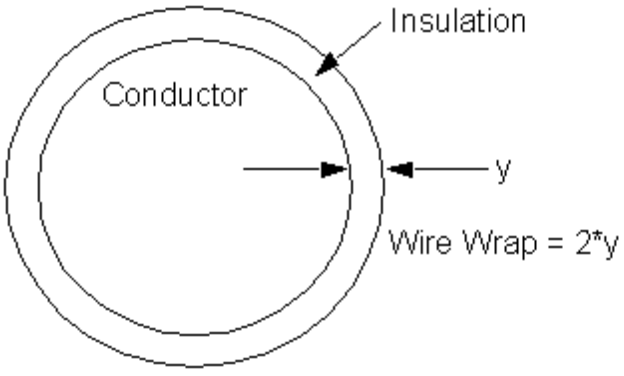
	
Sin_1	A first-class sinusoidal coil for double layer only. The **Conductors per Layer** field defines the maximum number of conductors in the slot. The software will determine the winding distribution in the slots to get the sinusoidal current distribution. 
Sin_2	A second-class sinusoidal coil for double layer only. The **Conductors per Layer** field defines the maximum number of conductors in the slot. The software will determine the winding distribution in the slots to get the sinusoidal current distribution. 

c. Once you have clicked a button to select a winding, click **OK** to close the **Winding Type** window and return to the **Properties** window.

10. Click the **Main (A)** tab.
    a. Enter the end length adjustment of the main stator coil in the **End Extension** field. The end adjustment is the distance one end of the conductor extends vertically beyond the end of the stator.



    b. Enter the number of conductors per layer of main winding in the **Conductors per Layer** field.
    c. Enter the number of parallel branches in the main stator winding in the **Parallel Branches** field.
    d. Enter the number of wires per conductor in the **Number of Strands** field. Enter 0 to have RMxprt auto-design the value.
    e. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



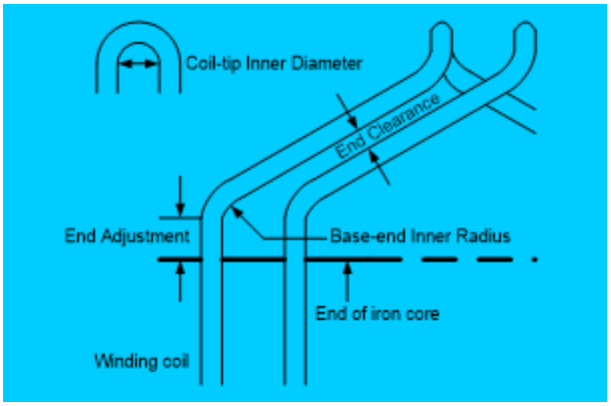
    f. To select the **Wire Size**: Click the button for **Wire Size**.
    The **Wire Size** window appears.

g. Select a value from the **Wire Diameter** pull-down list.

h. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

<*number*>	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.
USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.
AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.
MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21mm and 2 with a diameter of 0.13mm.

The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>General Options>Machines**command.



i. Enter the inner radius of the base corner in the **Base Inner Radius** field.

j. Enter the inner diameter of the coil tip in the **Tip Inner Diameter** field.

k. Enter the distance between two stator coils in the **End Clearance** field.

l. Enter the thickness of the slot liner insulation in the **Slot Liner** field.

m. Enter the thickness of the wedge insulation in the **Wedge Thickness** field.

n. Enter the thickness of the insulation layer in the **Layer Insulation** field.

o. Enter the limited slot fill factor for the wire design in the **Limited Fill Factor** field. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.

p. When you are done setting the wire size, click **OK** to close the **Wire Size** window and return to the **Properties** window.

11. Click the **Aux (B)** tab.

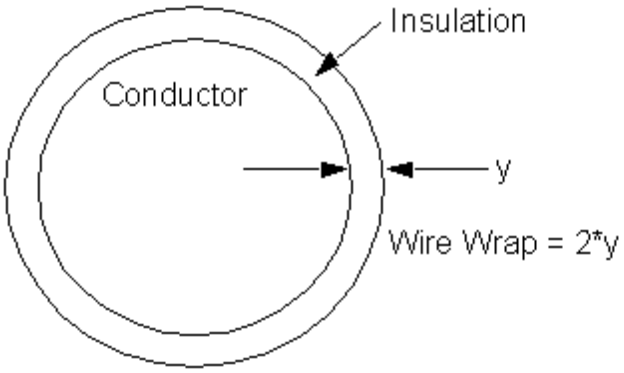
a. Enter the end length adjustment of the auxiliary stator coil in the **End Extension** field.

b. Enter the number of conductors per layer of auxiliary winding in the **Conductors per Layer** field.

c. Enter the number of parallel branches in the auxiliary stator winding in the **Parallel Branches** field.

d. Enter the number of wires per conductor in the **Number of Strands** field. Enter 0 to have RMxprt auto-design the value.

e. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



f. To select the **Wire Size**: Click the button for **Wire Size**.
The **Wire Size** window appears.

g. Select a value from the **Wire Diameter** pull-down list.

h. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

< ***number*****>**	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.
USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.

AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.
MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21mm and 2 with a diameter of 0.13mm.

The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>General Options>Machines**command.

i. When you are done setting the wire size, click **OK** to close the **Wire Size** window and return to the **Properties** window.

12. Click **OK** to close the **Properties** window.

Stator Winding Data for Single-Phase Induction Motors

To access the stator winding data, double-click the **Machine-Stator-Winding** entry in the project tree.

The **Stator Winding Data Properties** window contains the following fields:

Winding tab	**Slot Liner**	The thickness of the slot liner.
	Wedge Thickness	The thickness of the wedge insulation
	Limited Fill Factor	The limited slot fill factor for the wire design. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.
	Winding Layers	The number of winding layers. Not visible if the **Winding Type** is either **Sin_1** or **Sin_2**.
	Coil Pitch	The coil pitch measured in number of slots. Not visible if the **Winding Type** is either **Sin_1** or **Sin_2**.
	Winding Type	The type of stator winding for the main phase. Click the button to open the **Winding Type** window and choose from **Lap**, **Sin_1**, **Sin_2**,and **Editor**.
Main (A)	**End Extension**	The end length adjustment of the stator coils.
	Conductors per Layer	The number of conductors per layer in the main winding.
	Parallel Branches	The number of parallel branches in the main stator winding.

	Number of Strands	The number of wires per conductor (0 for auto-design).
	Wire Wrap	The thickness of the double-sided wire wrap (0 for auto-pickup from the wire library).
	Wire Size	The wire diameter (0 for auto-design).
Aux (B)	**End Extension**	The end length adjustment of the auxiliary winding.
	Conductors per Layer	The number of conductors per layer in the auxiliary winding.
	Parallel Branches	The number of parallel branches in the auxiliary stator winding.
	Number of Strands	The number of wires per conductor in the auxiliary winding (0 for auto-design).
	Wire Wrap	The thickness of the double-sided wire wrap (0 for auto-pickup from the wire library).
	Wire Size	The wire diameter (0 for auto-design).

Winding Editor for a Single-Phase Induction Motor

For a single-phase induction motor, you may want to specify a different number of conductors for each stator slot. The Winding Editor makes this possible by enabling you to specify the number of turns for each coil.

To specify the number of turns for each coil:

1. Click **Machine>Winding>Edit Layout**.
   The **Winding Editor** window appears.
2. In the table in the upper left, set which phase you want for each coil and which slot is the “in” and “out” slot for the current in each coil.
3. If you are working on a quarter or half model, you may want to specify a multiplier by clicking the **Periodic Multiplier** check box and specifying a value.
4. Select or deselect the **Constant Turns** or **Constant Pitch** check boxes, depending on whether you want to be able to change these setting in the table above. When these options are selected, you cannot change the turns or pitch.
5. When you are satisfied with the coil settings, click **OK** to close the **Winding Editor** window.

Defining Different Size Wires for a Single-Phase Induction Motor

Use the **Gauge** option if you have a conductor that is made up different size wires.

To define different size wires:

1. In the **Wire Size** window, select **MIXED** from the **Gauge** pull-down menu.
2. Select either **Round** or **Rectangular** as the **Wire Type**.

3. Enter the appropriate wire data in the table:
   - For a round wire:
     - Enter the **Diameter** in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this diameter.
   - For a rectangular wire:
     - Enter the **Width** of the wire in the table.
     - Enter the **Thickness** of the wire in the table.
     - Enter the **Fillet** value in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this data.
4. Choose **Add** to add the new wire data.
5. Repeat steps 3 and 4 for each size wire you want to add.
6. When you are finished defining the wires, click **OK** to close the **Wire Size** window.

Note	For example, if one conductor is made up of 5 wires, and 3 of those wires have a diameter of 0.21mm, and the other 2 have a diameter of 0.13mm, then the mixed wire size table will have two lines. The first line will list Diameter = 0.21 and Number = 3. The second line will list Diameter = 0.13 and Number = 2. An equivalent wire diameter is displayed as Wire Size value in the Winding tab in the Properties window.

Defining the Rotor Data for a Single-Phase Induction Motor

The rotor consists of copper bars in which current is induced from the stator windings. The rotor rotates at a slightly slower speed than the stator electromagnetic field. In the project tree, double-click **Machine>Rotor**, **Machine-Rotor-Slot**, and **Machine-Rotor-Winding** to define the physical dimensions, slot data, wires, and conductors for the rotor.

To define the general rotor data:

1. To open the **Rotor Properties** window, double-click the **Machine>Rotor** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the stacking factor for the rotor core in the **Stacking Factor** field.
3. Enter the **Number of Slots** in the rotor.
4. Select the **Slot Type**:
   a. Click the button for the **Slot Type**.
   The **Select Slot Type** window appears.
   b. Select a slot type (available types include 1 through 4).

Note	When you place the mouse cursor over the slot type, a schematic of the selected type appears, displaying the slot dimension variables.

   c. Click **OK** to close the **Select Slot Type** window and return to the **Properties** window.

5. Enter the outer diameter of the rotor in the **Outer Diameter** field.
6. Enter the inner diameter of the rotor in the **Inner Diameter** field.
7. Enter the length of the rotor core in the **Length** field.
8. Select a **Steel Type** for the rotor core:
   a. Click the button for **Steel Type**.
   The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
9. Enter the **Skew Width**, measured in rotor slot pitch. This value defines by how much the rotor bars are skewed.
10. Optionally, select **Cast Rotor** to allow the conductor to fill all the space available in the slot. Otherwise, RMxprt assumes the slot wedge that fixes the bars is filled with insulator material in a 3D/3D geometry model.
11. Click **OK** to close the **Properties** window.

Rotor Data for Single-Phase Induction Motors

To access the general rotor data, double-click the **Machine>Rotor** entry in the project tree.

The **Rotor Data Properties** window contains the following fields:

Stacking Factor	The stacking factor of the rotor core.
Number of Slots	The number of slots the rotor core contains.
Slot Type	The type of slots in the rotor core. Click the button to open the **Select Slot Type** window.
Outer Diameter	The outer diameter of the rotor core.
Inner Diameter	The inner diameter of the rotor core.
Length	The length of the rotor core.
Steel Type	The steel type of the rotor core. Click the button to open the **Select Definition** window.
Skew Width	The skew width measured in slot number.
Cast Rotor	Select or clear this to specify whether the rotor squirrel-cage winding is cast or not.

Defining the Rotor Slots for Single-Phase Induction Motors

To define the rotor's slots:

1. To open the **Rotor Slot Properties** window, double-click the **Machine-Rotor-Slot** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the slot dimensions in the following fields: **Hs0**, **Hs01**, **Hs2**, **Bs0**, **Bs1**, **Bs2, Rs**.
3. Click **OK** to close the **Properties** window.

Rotor Slot Data for Single-Phase Induction Motors

To access the rotor slot data, double-click the **Machine-Rotor-Slot** entry in the project tree.

The **Rotor Slot Data Properties** window contains the following fields:

Hs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Hs01	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Hs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Hs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Bs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Bs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Bs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Rs	A slot dimension. (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected). Rs is added when the slot type is 3 or 4.

Defining the Rotor Windings for Single-Phase Induction Motors

To define the rotor windings:

1. To open the **Rotor Winding Properties** window, double-click the **Machine-Rotor-Winding** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select a **Bar Conductor Type** for the rotor winding bar:
   a. Click the button for **Bar Conductor Type**.
   The **Select Definition** window appears.
   b. Select a conductor type from the list, or define a new conductor type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
3. Enter the length of the gap between the end ring and the iron core in the **End Length** field. This field specifies the value for only one end of the gap, not both.

4. Enter the end ring dimension in the axial direction in the **End-Ring Width** field. The end ring connects the bars of the rotor to one another.
5. Enter the end ring dimension in the radius direction in the **End-Ring Height** field. The end ring's height covers at least the cross-section of the rotor conductor.
6. Select an **End Ring Conductor Type** for the rotor winding end ring:
   a. Click the button for **End Ring Conductor Type**.
   The **Select Definition** window appears.
   b. Select a conductor type from the list, or define a new conductor type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
7. Click **OK** to close the **Properties** window.

Rotor Winding Data for Single-Phase Induction Motors

To access the rotor winding data, double-click the **Machine-Rotor-Winding** entry in the project tree.

The **Rotor Winding Data Properties** window contains the following fields:

Bar Conductor Type	The type of bar conductor used in the winding. Click the button to open the **Select Definition** window.
End Length	The length of the single-side end of the extended bar.
End Ring Width	The width of one side of the end rings in the axial direction.
End Ring Height	The height of the end rings in the radian direction.
End Ring Conductor Type	The type of end ring conductor used in the winding. Click the button to open the **Select Definition** window.

Defining the Shaft Data for a Single-Phase Induction Motor

To define the shaft:

1. To open the **Shaft Data Properties** window, double-click the **Machine>Shaft** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select or clear the **Magnetic Shaft** check box to specify whether or not the shaft is to be made of magnetic material.
3. Click **OK** to close the **Properties** window.

Shaft Data for Single-Phase Induction Motors

To access the shaft data, double-click the **Machine>Shaft** entry in the project tree.

The **Shaft Data Properties** window contains the following fields:

Magnetic Shaft	Select or clear this check box to indicate whether or not the shaft is made of magnetic material.

Adding or Removing a Vent from a Single-Phase Induction Motor

By option, you can add a vent to a single-phase induction motor. To add a vent:.

1. Select the rotor icon in the project tree.
2. Right-click to display the pop-up menu and select **Insert Vent**.
   The vent icon appears in the project tree under the rotor.

To remove a vent from a rotor in a three-phase induction motor.

1. Select the rotor icon in the project tree.
2. Right-click to display the pop-up menu and select **Remove Vent**.
   The vent icon disappears in the project tree under the rotor.

The Vent data for the stator includes the following fields.

Holes per row	Number of axial vent holes per row.
Inner hole diameter	Diameter of vent holes in inner row.
Outer hole diameter	Diameter of vent holes in outer row.
Inner hole location	Center to center diameter of inner hole vents
Outer hole location	Center to center diameter of outer hole vents.

Setting Up Analysis Parameters for a Single-Phase Induction Motor

To define the solution data:

1. To open the **Solution Setup** window, right-click **Analysis** in the project tree, and click **Add Solution Setup**.
2. Click the **General** tab. The **Operation Type** is automatically set to **Motor** for this machine type.
3. Select the **Load Type** used in the motor from the following options:

Const Speed	The speed remains constant in the motor.
Const Power	The output power remains constant in the motor.
Const Torque	The torque remains constant regardless of the speed. In this case, $T_{load} = T_{rated}$, given by the output power divided by the given rated speed.
Linear Torque	The torque increases linearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})$ where T_{rated} is given by the output power divided by the given rated speed.

Fan Load	The load varies nonlinearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})^2$ where T_{rated} is given by the output power divided by the given rated speed.

4. Enter the output power developed at the shaft of the motor in the **Rated Output Power** field.
5. Enter the RMS line-to-line voltage in the **Rated Voltage** field.
6. Enter the desired output speed of the motor at the load point in the **Rated Speed** field.
7. Enter the temperature at which the system functions in the **Operating Temperature** field.
8. Click the **Single-Phase Induction Motor** tab.
9. Enter the electrical line frequency in the **Frequency** field, and select the units.
10. Click **OK** to close the **Solution Setup** window.

Related Topics

Solution Data for Single-Phase Induction Motors

Solution Data for Single-Phase Induction Motors

To access the solution data, right-click **Analysis** in the project tree, and click **Add Solution Setup**.

The **Solution Setup** window contains the following fields:

Operation Type	*On the* **General** *tab.* The operation type is automatically set to **Motor** for this machine type.
Load Type	*On the* **General** *tab.* Select from **Const Speed**, **Const Power**, **Const Torque**, **Linear Torque**, and **Fan Load**. The default is **Const Power**.
Rated Output Power	*On the* **General** *tab.* Type a value for the rated output voltage, and select the units.
Rated Voltage	*On the* **General** *tab.* Type a value for the rated voltage, and select the units.
Rated Speed	*On the* **General** *tab.* Type a value for the rated speed, and select the units.
Operating Temperature	*On the* **General** *tab.* Type a value for the operating temperature, and select the units.
Frequency	*On the* **Single-Phase Induction Motor** *tab.* Type a value for the frequency, and select the units.

Related Topics

Setting Up Analysis Parameters for a Single-Phase Induction Motor

Adjust-Speed Synchronous Machines

After you have selected **Adjust-Speed Synchronous Machines** as your model type, you need to define the following:

- General data, such as the voltage, speed, and circuit type of the model.
- Circuit data, such as trigger pulse width, transistor drop, and control circuit information.
- Stator data, such as the diameter, slot dimensions, and skew width of the stator.
- Stator Winding
- Rotor pole data, such as the associated permanent-magnet dimensions, air gap, and stacking factor.
- Solution data, such as rated output voltage and frequency.

By option, you can:

- Add a machine housing.

Analysis Approach for Adjust-Speed Synchronous Machines

In adjustable-speed permanent-magnet synchronous machines, the rotor speed is controlled by adjusting the frequency of the input voltage. Unlike standard brushless permanent-magnet DC motors, this type of machine does not utilize position sensors.

Permanent magnets are mounted on the rotor of a permanent-magnet synchronous machine, which could be either inner or outer rotor type. The poly-phase armature winding is embedded in the stator, whose number of poles is the same as that of the rotor. The machine can operate as a generator or as a motor. When the machine operates as a motor, the stator poly-phase winding can be fed either by a sinusoidal AC source or by a DC source via a DC to AC inverter. When the machine operates as a generator, the stator poly-phase winding supplies an AC source for electric loads.

Stator Winding Connected to a Sinusoidal AC Source

In this case, the performance of the machine can be analyzed in the frequency domain based on the phasor diagrams, as shown in Figure 6.1 for the generators and Figure 6.2 for the motors.

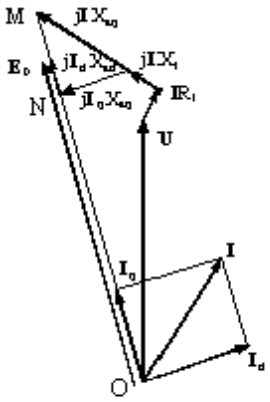


Figure 6.1 The phasor diagram for generators

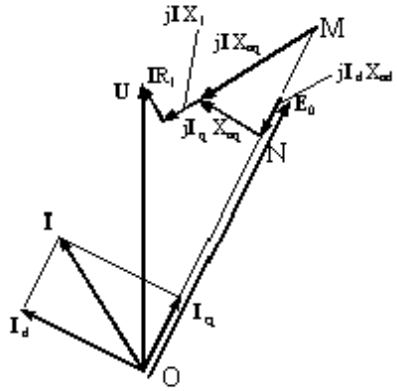


Figure 6.2 The phasor diagram for motors

In the figures, R_1 and X_1 are the resistance and the leakage reactance of the armature winding, X_{ad} and X_{aq} are the d-axis armature reactance and the q-axis armature reactance, respectively. In the phasor diagram, X_{ad} is a linearized nonlinear parameter, and X_{aq} is a linear parameter. The d-axis synchronous reactance X_d and q-axis synchronous reactance X_{aq} are calculated directly from

Let ? denote the power angle for a generator (the angle that U lags E_0), or the torque angle for a motor (the angle that E_0 lags U), then we have

$$X_d = X_1 + X_{ad}$$

$$X_q = X_1 + X_{aq}$$

$$I_d X_d + I_q R_1 = \pm(U\cos\theta - E_0)$$

$$-I_d R_1 + I_q X_q = U\sin\theta$$

where the plus sign + is for the motor and the minus sign - is for the generator.
Solving for I_d and I_q yields

$$I_d = \frac{\pm X_q(U\cos\theta - E_0) - R_1 U\sin\theta}{R^2_1 + X_d X_q}$$

$$I_q = \frac{\pm R_1(U\cos\theta - E_0) - X_d U\sin\theta}{R^2_1 + X_d X_q}$$

where the plus sign + is for the motor and the minus sign - is for the generator.

Let the angle that *I* lags E_0 be ψ, we have

$$\psi = \tan^{-1}\frac{I_d}{I_q}$$

The power factor angle φ (the angle that *I* lags *U*) is

$$\varphi = \psi \pm \theta$$

where the plus sign + is for the motor and the minus sign - is for the generator.

For the motor operation, the input electric power is

$$P_1 = 3UI\cos\varphi$$

The output mechanical power is

$$P_2 = P_1 - (P_{fw} + P_{Cua} + P_{Fe})$$

where P_{fw}, P_{Cua}, and P_{Fe} denote the frictional and wind, the armature copper and the iron-core losses, respectively.

The output mechanical torque is

$$T_2 = \frac{P_2}{\omega}$$

where ω denotes the synchronous speed in rad/s.

The efficiency of the motor is

$$\eta = \frac{P_2}{P_1} \times 100\,\%$$

For the generator operation, the output electric power is

$$P_2 = 3UI\cos\varphi$$

The input mechanical power is

$$P_1 = P_2 + P_{fw} + P_{Cua} + P_{Fe}$$

where P_{fw}, P_{Cua}, and P_{Fe} denote the frictional and wind, the armature copper and the iron-core losses, respectively.

The input mechanical torque is

$$T_1 = \frac{P_1}{\omega}$$

where ω denotes the synchronous speed in mechanical rad/s.

The efficiency of the generator is

$$\eta = \frac{P_2}{P_1} \times 100\ \%$$

Stator Winding Fed by a DC to AC Inverter

In this case, this adjustable-speed synchronous machine (ASSM) operates as a motor, and the analysis approach is similar to that of a brushless DC (BLDC) motor. The stator poly-phase armature winding is connected to a DC power supply through a DC to AC inverter to produce the rotational magnetic field in the air-gap. The main difference between ASSM and BLDC motor is: in BLDC motor, trigger time exactly **depends on** the rotor position; but in ASSM, the trigger time is **independent of** the rotor position. If the mechanical load of a BLDC motor increases, the rotor speed and the induced voltage decreases, causing the armature current and torque increase to balance the increased mechanical load. However, for an ASSM, if the mechanical load increases, the rotor speed decreases temporarily, which causes the torque angle (the same as lead angle of trigger for a BLDC motor) increase and then torque increase to retain the synchronous speed.

Therefore, the speed of a BLDC motor varies with input voltage and mechanical load, while the speed of an ASSM does not. The speed of an ASSM can be changed by adjusting the frequency of the controlling signal, which explains why it is called *Adjustable-Speed Permanent-Magnet Synchronous Machine*.

Using the time-domain mathematical model to analyze the characteristics of the electric machine, Park's voltage equation in the matrix form is as follows

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} - \begin{bmatrix} e_d \\ e_q \\ e_0 \end{bmatrix} = \begin{bmatrix} R_1 + L_d\rho & -L_q\omega_e & 0 \\ -L_d\omega_e & R_1 + L_q\rho & 0 \\ 0 & 0 & R_1 + L_0\rho \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix}$$

where R_1 is the armature winding resistance, L_d, L_q and L_0 are the d-, the q- and the 0-axis inductances respectively, ω_e is the revolution speed in electric radians per second, the differential operator is

$$\rho = \frac{\mathrm{d}}{\mathrm{d}t}$$

The coordinate transformation equations for the terminal voltage, the induced emf and the armature winding current are

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = C^T \begin{bmatrix} v_a \\ v_b \\ \vdots \end{bmatrix} \qquad \begin{bmatrix} e_d \\ e_q \\ e_0 \end{bmatrix} = C^T \begin{bmatrix} e_a \\ e_b \\ \vdots \end{bmatrix} \qquad \begin{bmatrix} i_a \\ i_b \\ \vdots \end{bmatrix} = C \begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix}$$

The transformation matrices for the two-, the three- and the four-phase systems are C_2, C_3 and C_4, respectively, as follows

$$C_2 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \end{bmatrix}$$

$$C_3 = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos\theta & \sin\theta & \frac{1}{\sqrt{2}} \\ \cos(\theta-\alpha) & \sin(\theta-\alpha) & \frac{1}{\sqrt{2}} \\ \cos(\theta-2\alpha) & \sin(\theta-2\alpha) & \frac{1}{\sqrt{2}} \end{bmatrix}$$

$$C_4 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ \sin\theta & -\cos\theta & 0 \\ -\cos\theta & -\sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \end{bmatrix}$$

where

$$\alpha = \frac{2}{3}\pi$$

The input electric power is obtained from the voltage and the current as:

$$P_1 = \frac{1}{T}\int_0^T (v_d i_d + v_q i_q + v_0 i_0)dt$$

The output mechanical power is:

$$P_2 = P_1 - (P_{fw} + P_{Cua} + P_{Fe})$$

where P_{fw}, P_{Cua}, P_t and P_{Fe} denote the frictional and wind, the armature copper, the switching and the iron-core losses, respectively.

The output mechanical torque is

$$T_2 = \frac{P_2}{\omega}$$

where ω denotes the revolution speed in mechanical radians per second.

The efficiency of the electric machine is

$$\eta = \frac{P_2}{P_1} \times 100\ \%$$

Defining an Adjustable-Speed Synchronous Machine

The general procedure for defining a adjust-speed synchronous machine is as follows:

1. Insert the adjust-speed synchronous machine into a new or existing project.
2. Double-click the **Machine** entry in the project tree to define the general data.
3. Double-click the **Machine>Circuit** entry in the project tree to define the control circuit.
4. Double-click the **Machine>Stator** entry in the project tree to define the stator geometry.
5. Double-click the **Machine-Stator-Slot** entry in the project tree to define the stator slot dimensions.
6. Double-click the **Machine-Stator-Winding** entry in the project tree to define the stator windings and conductors.
7. Double-click the **Machine-Rotor** entry in the project tree to define the rotor geometry.
8. Double-click the **Machine-Rotor-Pole** entry in the project tree to define the pole, embrace, offset, and air gap data for the rotor pole.
9. Double-click the **Machine>Shaft** entry in the project tree to define the magnetism of the shaft.
10. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
11. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
12. Choose **File>Save** to save the project.
13. Choose **RMxprt>Analyze** to analyze the design.

Note	When you place the cursor over an entry field in the data windows, a brief description of that field appears in the status bar at the bottom of the **RMxprt** window.

Once analyzed, the model can be viewed in the Maxwell 2D Modeler, or it can be used to create a new Maxwell 2D project, and a new Maxwell 3D design.

Defining the General Data for an Adjust-Speed Synchronous Machine

Use the **General** window to define the basic parameters of the motor, such as the motor's rated output power, rated voltage, losses, and circuit types.

To define the general data:

1. To open the **General Data Properties** window, double-click the **Machine** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the number of poles for the machine in the **Number of Poles** field. This value is the total number of poles in the stator (or the number of pole pairs multiplied by two).
3. Enter the energy loss due to friction at the given speed in the **Frictional Loss** field.
4. Enter the wind loss due to air resistance measured at the reference speed in the **Windage Loss** field.
5. Enter the given speed in the **Reference Speed** field.
6. Select one of the following from the **Control Type** pull-down list:
   - **DC**: Switched DC voltage at the given input frequency.
   - **PWM**: Pulse width modulation. When you select this source type, you must enter the following values in the **Circuit Data Properties** window: **Modulation Index** (the ratio of the sine wave amplitude to the triangular amplitude) and **Carrier Frequency Times** (the ratio of the triangular frequency to the sine wave frequency).
   - **AC**: An AC excitation.
7. Select a **Circuit Type** from the following types:

Y3	Y-connected, three-phase.
L3	Loop-type, three-phase.
S3	Star-type, three-phase.
C2	Cross-type, two-phase.
L4	Loop-type, four-phase.
S4	Star-type, four-phase.

   The circuit types are based on industry standards. By default, type **Y3**, a three-phase, six-status circuit, is selected as the circuit type.

Note	When you place the mouse cursor over a circuit type, an outline schematic of the circuit appears.

8. Click **OK** to close the **Properties** window.

General Data for Adjust-Speed Synchronous Machines

To access the general data, double-click the **Machine** entry in the project tree.

The **General Data Properties** window for a three-phase induction motor contains the following fields:

Machine Type	The machine type you selected when inserting a new RMxprt design (Adjust-Speed Synchronous Machine).
Number of Poles	The number of poles the machine contains.
Rotor Position	Select whether the rotor is an **Inner Rotor** or **Outer Rotor**.
Frictional Loss	The frictional energy loss (due to friction) measured at the reference speed.
Windage Loss	The windage loss (due to air resistance) measured at the reference speed.
Reference Speed	The given speed of reference.
Control Type	The way the circuit is controlled. Select from **DC**, **PWM** (pulse-width modulation), or **AC**.
Circuit Type	The drive circuit type. Click the button to open the **Circuit Type** window and select from the following six types: • **Y3**: Y-Type, 3-Phase • **L3**: Loop-Type, 3-Phase • **S3**: Star-Type, 3-Phase • **C2**: Cross-Type, 2-Phase • **L4**: Loop-Type, 4-Phase • **S4**: Star-Type, 4-Phase

Defining the Circuit Data for an Adjust-Speed Synchronous Machine

Use the **Circuit Data Properties** window to define the circuit properties for an adjustable-speed synchronous machine.

Note	No circuit data properties exist when **AC** is selected as the **Control Type**.

1. To open the **Circuit Data Properties** window, double-click the **Machine>Circuit** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. If you selected DC as the **Control Type**, enter the period from on-status to off-status of a transistor, in electrical degrees, in the **Trigger Pulse Width** field.

3. Enter the voltage drop across one transistor when the transistor is turned on in the **Transistor Drop** field. Refer to the figures of the different circuit types in step 2.
4. Enter the voltage drop of one diode in the discharge loop in the **Diode Drop** field. If you selected a star-type circuit (**S3** or **S4**) as the **Circuit Type**, enter the total discharge voltage in this field.
5. If you selected PWM as the **Control Type**, then enter values in the following two fields:
   - **Modulation Index**: The ratio of the sine-wave amplitude to the triangular amplitude. (PWM circuits only.)
   - **Carrier Frequency Times**: The ratio of the triangular frequency to the sine-wave frequency. (PWM circuits only.)
6. Click **OK** to close the **Properties** window.

Circuit Data for Adjust-Speed Synchronous Machines

To access the **Circuit Data Properties** window, double-click the **Machine>Circuit** entry in the project tree. No circuit data properties exist when **AC** is selected as the **Control Type**.

Trigger Pulse Width	The period from on-status to off-status for a transistor, in electrical degrees. (DC circuits only.)
Transistor Drop	The voltage drop across one transistor when the transistor is turned on.
Diode Drop	The voltage drop across one diode in the discharge loop.
Modulation Index	The ratio of the sine-wave amplitude to the triangular amplitude. (PWM circuits only.)
Carrier Frequency Times	The ratio of the triangular frequency to the sine-wave frequency. (PWM circuits only.)

Defining the Stator Data for an Adjust-Speed Synchronous Machine

Use the **Stator Properties** windows to define the stator dimensions, slots, windings, and conductors.

The stator is the outer lamination stack where the polyphase voltage windings reside.

To define the general stator data:

1. To open the **Stator Data Properties** window, double-click the **Machine>Stator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the **Outer Diameter** of the stator.
3. Enter the **Inner Diameter** of the stator.
4. Enter the length of the stator core in the **Length** field.

5. Enter the stacking factor for the stator core in the **Stacking Factor** field.
6. Select a **Steel Type** for the stator core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
7. Enter the **Number of Slots** in the stator.
8. Select the **Slot Type**:
   a. Click the button for the **Slot Type**.
      The **Select Slot Type** window appears.
   b. Select a slot type (available types include 1 through 4).

Note	When you place the mouse cursor over the slot type, a schematic of the selected type appears, displaying the slot dimension variables.

   c. Click **OK** to close the **Select Slot Type** window and return to the **Properties** window.
9. Enter the skew width, measured in slot number, in the **Skew Width** field.
10. Click **OK** to close the **Properties** window.

Defining the Stator Dimensions and Slots

To define the stator slots:

1. To open the **Stator Slot Data Properties** window, double-click the **Machine-Stator-Slot** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Optionally, to automatically design the dimensions of slots **Hs2**, **Bs1**, and **Bs2**, select the **Auto Design** check box.
3. Optionally, to design dimensions of slots **Bs1** and **Bs2** based on the stator tooth width, select the **Parallel Tooth** check box, and enter a value in the **Tooth Width** field.

4. Enter the available slot dimensions.

Hs0	Always available.
Hs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.
Bs0	Always available.
Bs1	Available only when **Auto Design** and **Parallel Tooth** are both cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Bs2	Available only when **Auto Design** and **Parallel Tooth** are both cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Rs	Rs is added when the slot type is 3 or 4.

5. Click **OK** to close the **Properties** window.

Stator Data for Adjust-Speed Synchronous Machines

To access the general stator data, double-click the **Machine>Stator** entry in the project tree.

The **Stator Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the stator core.
Inner Diameter	The inner diameter of the stator core.
Length	The length of the stator core.
Stacking Factor	The stacking factor of the stator core.
Steel Type	The steel type of the stator core. Click the button to open the **Select Definition** window.
Number of Slots	The number of slots the stator core contains.
Slot Type	The type of slots in the stator core. Click the button to open the **Select Slot Type** window.
Skew Width	The skew width measured in slot number.

Stator Slot Data for Adjust-Speed Synchronous Machines

To access the stator slot data, double-click the **Machine-Stator-Slot** entry in the project tree.

The **Stator Slot Data Properties** window contains the following fields:

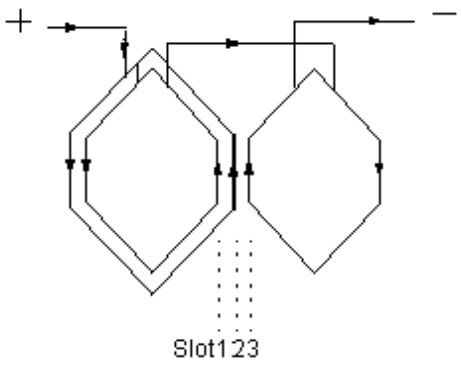
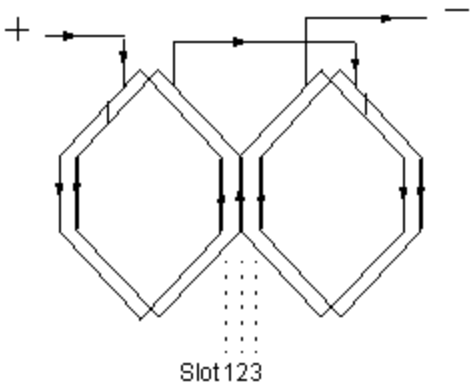
Auto Design	Select or clear this to enable or disable auto-design of slots **Hs2**, **Bs1**, and **Bs2**. When this check box is selected, only two other fields appear in the window: **Hs0** and **Bs0**.
Parallel Tooth	Select this to design **Bs1** and **Bs2** based on the tooth width. When this check box is selected, the **Bs1** and **Bs2** fields are removed, and the **Tooth Width** field is added.
Tooth Width	The tooth width for the parallel tooth, on which **Bs1** and **Bs2** are designed.
Hs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Hs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Hs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Rs	A slot dimension. (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected). Rs is added when the slot type is 3 or 4.

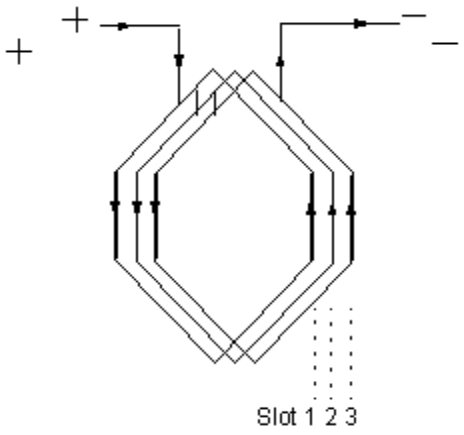
Defining the Stator Windings and Conductors for an Adjust-Speed Synchronous Machine

To define the stator windings and conductors:

1. To open the **Stator Slot Winding Properties** window, double-click the **Machine-Stator-Winding** entry in the project tree on the desktop. You can also enter values in the **Properties** section of the desktop without opening a separate window.
2. Click the **Winding** tab.
3. Enter the number of layers in the stator winding in the **Winding Layers** field.
4. Select a **Winding Type**:
   a. Click the button for **Winding Type**.
   The **Winding Type** window appears.
   b. Select from one of the following three types of winding:
      - **Whole Coiled**
      - **Half Coiled**
      - **Editor**

When you place the mouse cursor over a winding button, an outline of the selected winding appears. The following table describes the six types of windings that are possible (three for one-layer and three for two-layer):

Type	Description
One Layer Winding Editor	A user-defined one-layer winding arrangement. You need to set up the winding arrangement for each slot.
Whole Coiled	A one-layer whole-coiled winding: 
Whole Coiled	A two-layer wave winding:  The phase belt for this winding configuration is equal to 360/2*m*, where *m* is the phase number.
Half Coiled	A two-layer half-coiled winding:

	 There is only one coil per phase per pair of poles.

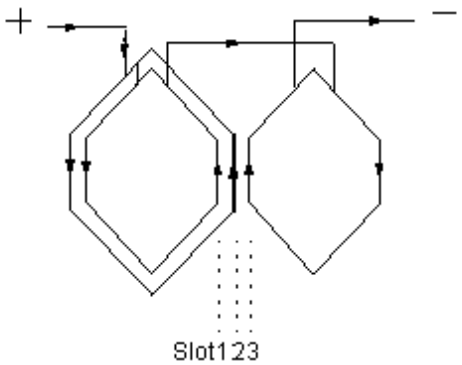
Note	**For a two-layer winding**, if you click **Constant Pitch** in the **Winding Editor**, only the top layer needs to be defined; the bottom layer will be determined according to the coil pitch.

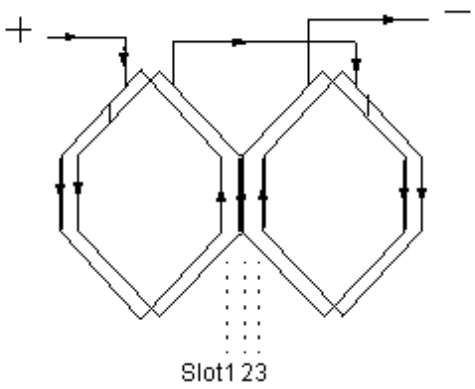
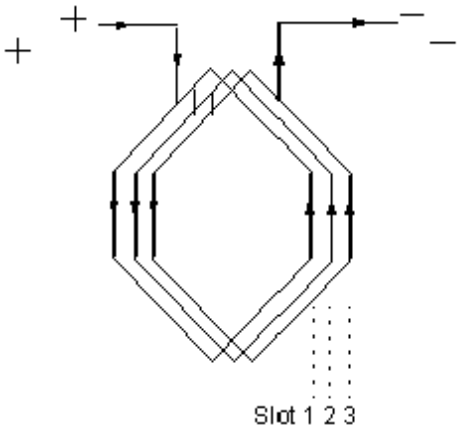
Once you have clicked a button to select a winding, click **OK** to close the **Winding Type** window and return to the **Properties** window. Select the **Winding Type** for the stator.

Note	When you place the mouse cursor over the winding type, a schematic of the selected winding appears.

Winding types **10** and **20** are user-defined. If you select either of these, a window appears, asking you to define the name of the winding arrangement. The window closes when the user-defined winding is entered.

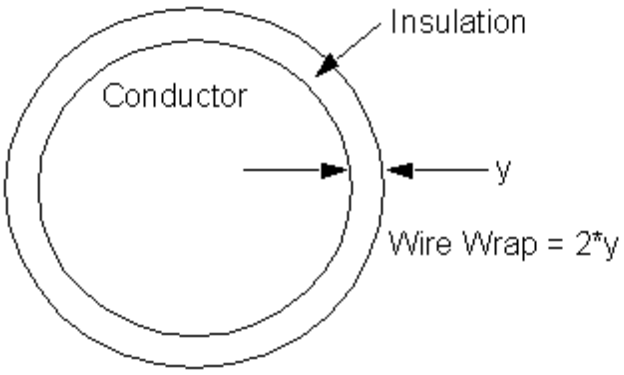
Select from the following winding types:

One-Layer Winding Editor	
	A user-defined single-layer winding arrangement. When you select this type, enter the winding arrangement, and choose **OK**.
11	A one-layer whole-coiled winding: 

20	A user-defined winding arrangement. When you select this type, enter the winding arrangement, and choose **OK**.
21	A two-layer wave winding:  The phase belt for this winding configuration is equal to 360/2*m*, where *m* is the phase number.
22	A two-layer winding: 

5. Select or enter the number of parallel branches in one phase of the winding in the **Parallel Branches** field.
6. Enter the total number of conductors in each stator slot in the **Conductors per Slot** field. This value is the number of turns per coil multiplied by the number of layers.
7. Enter the coil pitch, measured in number of slots, in the **Coil Pitch** field. The coil pitch is the number of slots separating one winding. For example, if a coil starts in slot 1 and ends in slot 6, it has a coil pitch of 5.

8. Enter the number of wires per conductor in the **Number of Strands** field. Enter 0 to have RMxprt auto-design this value.
9. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



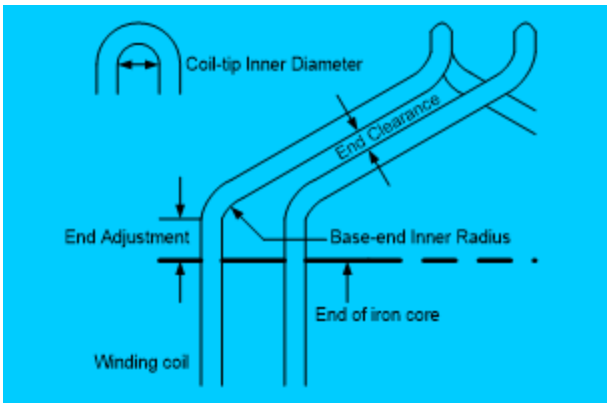
10. Select the **Wire Size**:
    a. Click the button for **Wire Size**.
    The **Wire Size** window appears.
    b. Select a value from the **Wire Diameter** pull-down list.
    c. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

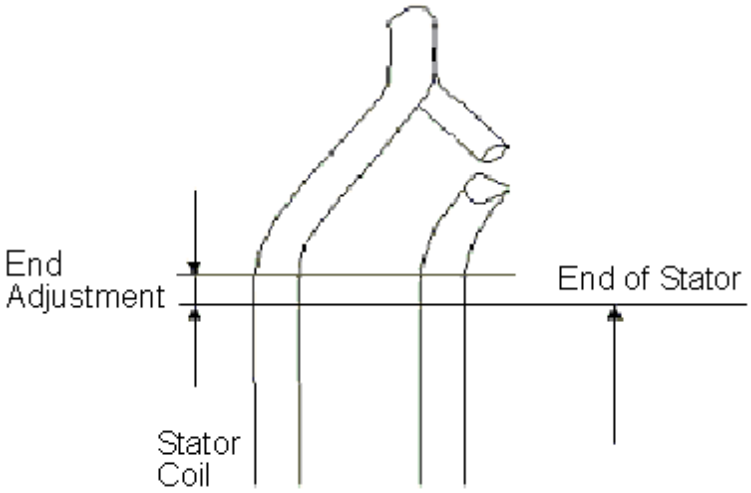
< ***number*** **>**	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.
USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.
AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.
MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21mm and 2 with a diameter of 0.13mm.

The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>General Options>Machines**command. When you are done setting the wire size, click **OK** to close the **Wire Size** window and return to the **Properties** window.

11. Click the **End/Insulation** tab.
12. Select or clear the **Input Half-turn Length** check box.
13. Do one of the following:

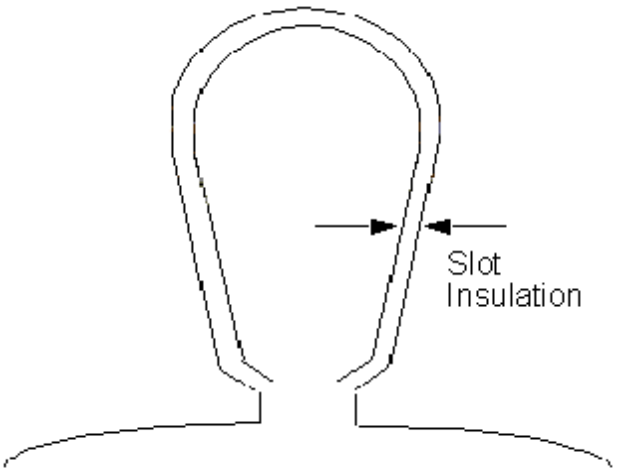
- If you selected **Input Half-turn Length**, then enter the half-turn length of the armature winding in the Half Turn Length field.
- If you cleared **Input Half-turn Length**, then enter the end length adjustment of the stator coils in the **End Adjustment** field. The end adjustment is the distance one end of the conductor extends vertically beyond the end of the stator.





14. Enter the inner radius of the base corner in the **Base Inner Radius** field.
15. Enter the inner diameter of the coil tip in the **Tip Inner Diameter** field.
16. Enter the distance between two stator coils in the **End Clearance** field.

17. Enter the thickness of the slot liner insulation in the **Slot Liner** field.



18. Enter the thickness of the wedge insulation in the **Wedge Thickness** field.
19. Enter the thickness of the insulation layer in the **Layer Insulation** field.
20. Enter the limited slot fill factor for the wire design in the **Limited Fill Factor** field. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.
21. Click **OK** to close the **Properties** window.

Winding Editor for an Adjustable-Speed Synchronous Machine

For a adjustable-speed synchronous machine, you may want to specify a different number of conductors for each stator slot. The Winding Editor makes this possible by enabling you to specify the number of turns for each coil.

To specify the number of turns for each coil:

1. Click **Machine>Winding>Edit Layout**.
   The **Winding Editor** window appears.
2. In the table in the upper left, set which phase you want for each coil and which slot is the “in” and “out” slot for the current in each coil.
3. If you are working on a quarter or half model, you may want to specify a multiplier by clicking the **Periodic Multiplier** check box and specifying a value.
4. Select or deselect the **Constant Turns** or **Constant Pitch** check boxes, depending on whether you want to be able to change these setting in the table above. When these options are selected, you cannot change the turns or pitch.
5. When you are satisfied with the coil settings, click **OK** to close the **Winding Editor** window.

Defining Different Size Wires for an Adjustable Speed Synchronous Machine

Use the **Gauge** option in the **Wire Size** dialog if you have a conductor that is made up different size wires.

To define different size wires:

1. In the **Wire Size** window, select **MIXED** from the **Gauge** pull-down menu.
2. Select either **Round** or **Rectangular** as the **Wire Type**.
3. Enter the appropriate wire data in the table:
   - For a round wire:
     - Enter the **Diameter** in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this diameter.
   - For a rectangular wire:
     - Enter the **Width** of the wire in the table.
     - Enter the **Thickness** of the wire in the table.
     - Enter the **Fillet** value in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this data.
4. Click **Add** to add the new wire data.
5. Repeat steps 3 and 4 for each size wire you want to add.
6. When you are finished defining the wires, click **OK** to close the **Wire Size** window and return to the RMxprt **Properties** window.

Note	For example, if one conductor is made up of 5 wires, and 3 of those wires have a diameter of 0.21mm, and the other 2 have a diameter of 0.13mm, then the mixed wire size table will have two lines. The first line will list Diameter = 0.21 and Number = 3. The second line will list Diameter = 0.13 and Number = 2. An equivalent wire diameter is displayed as Wire Size value in the Winding tab in the Properties window.

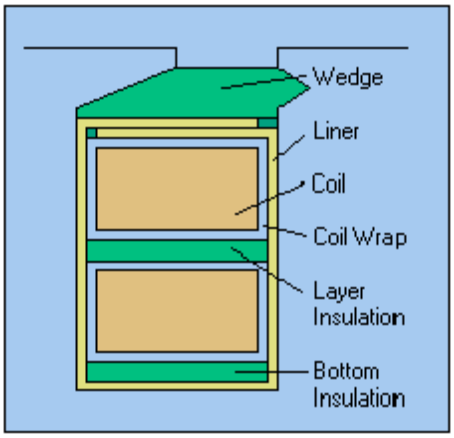
Stator Winding Data for Adjust-Speed Synchronous Machines

To access the stator winding data, double-click the **Machine-Stator-Winding** entry in the project tree.

The **Stator Winding Data Properties** window contains the following fields:

Winding tab	**Winding Layers**	The number of winding layers.
	Winding Type	The type of stator winding. Click the button to open the **Winding Type** window and choose from **Whole Coiled**, **Half Coiled**, and **Editor**.
	Parallel Branches	The number of parallel branches in the stator winding.

	Conductors per Slot	The number of conductors per stator slot (0 for auto-design).
	Coil Pitch	The coil pitch measured in number of slots.
	Number of Strands	The number of wires per conductor (0 for auto-design).
	Wire Wrap	The thickness of the double-sided wire wrap (0 to automatically obtain this value from the wire library).
	Wire Size	The diameter of the wire (0 for auto-design). Click the button to open the **Wire Size** window where you can specify units, wire type, diameter, and gauge.
End/Insulation tab	**Input Half-turn Length**	Select or clear this check box to specify whether or not you want to enter the half-turn length. When this check box is selected, the **Half Turn Length** field appears the next time you open the **Properties** window. When this check box is selected, the **End Adjustment** field appears instead.
	Half Turn Length	The half-turn length of the armature winding.
	End Adjustment	The end length adjustment of the stator coils, which is the distance one end of the conductor extends vertically beyond the end of the stator.
	Base Inner Radius	The inner radius of the base corner.
	Tip Inner Diameter	The inner diameter of the coil tip.
	End Clearance	The end clearance between two adjacent coils.
	Slot Liner	The thickness of the slot liner insulation.
	Wedge Thickness	The thickness of the wedge insulation.
	Layer Insulation	The thickness of the insulation layer.
	Limited Fill Factor	The limited slot fill factor for the wire design. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.



Defining the Rotor Data for an Adjust-Speed Synchronous Machine

The rotor consists of copper bars in which current is induced by the magnetic fields produced by the stator windings. In the project tree, double-click **Machine>Rotor** and **Machine-Rotor-Pole** to define the rotor and the pole.

To define general stator data:

1. To open the **Rotor Data Properties** window, double-click the **Machine>Rotor** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the outer diameter of the rotor in the **Outer Diameter** field.
3. Enter the inner diameter of the rotor in the **Inner Diameter** field.
4. Enter the length of the rotor core in the **Length** field.
5. Select a **Steel Type** for the rotor core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
6. Enter the stacking factor for the rotor core in the **Stacking Factor** field.
7. Select a **Pole Type**:
   a. Click the button.
      The **Select Pole Type** window appears.
   b. Click a button to select the desired pole type (1, 2, 3, 4, or 5). TIP: When you run the mouse over each option, the diagram changes to show that pole type.

Note	When you place the mouse cursor over a rotor type, an outline of the selected circuit type appears.

   c. Click **OK** to close the **Select Pole Type** window and return to the **Properties** window.

8. Click **OK** to close the **Properties** window.

Rotor Data for Adjust-Speed Synchronous Machines

To access the general rotor data, double-click the **Machine>Rotor** entry in the project tree.

The **Rotor Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the rotor core.
Inner Diameter	The inner diameter of the rotor core.
Length	The length of the rotor core.
Steel Type	The steel type of the rotor core. Click the button to open the **Select Definition** window.
Stacking Factor	The stacking factor of the rotor core.
Pole Type	The pole type for the rotor. Click this button to open the **Select Pole Type** window and select from the following types: **1**, **2**, **3**, **4**, *5*.

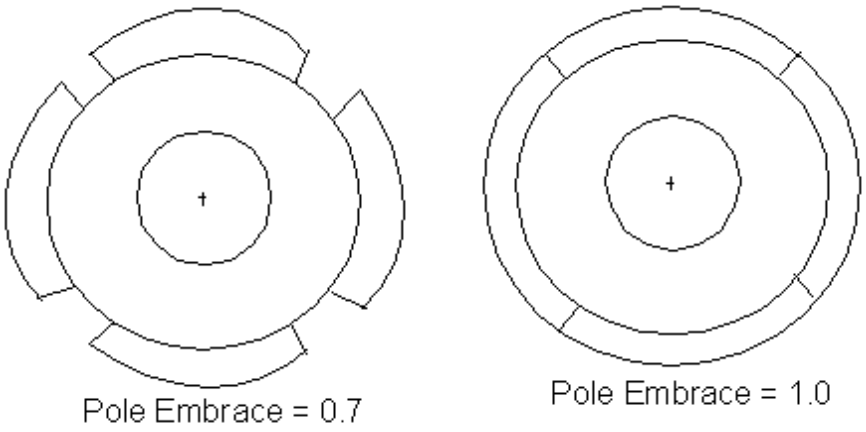
Defining the Rotor Pole for an Adjust-Speed Synchronous Machine

The rotor pole drives the electromagnetic field which is coupled with the stator windings. Use the **Rotor Pole Data Properties** window to define the rotor pole.

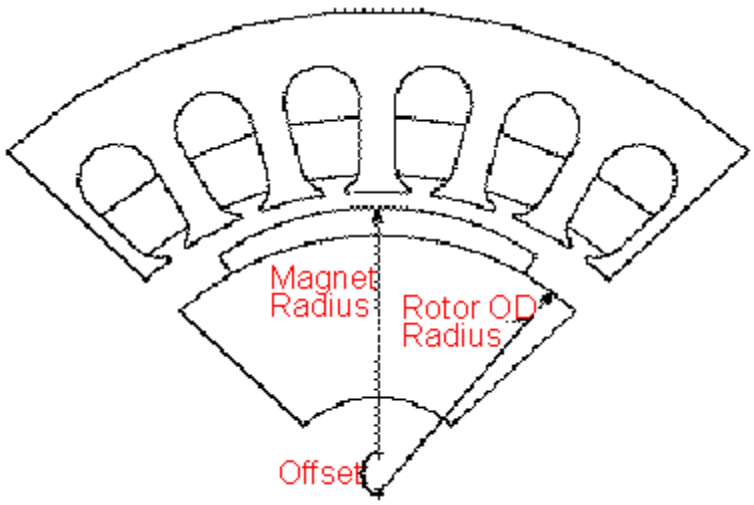
Note	Some of the fields in the **Rotor Pole** window change, or are inactive, depending on the **Rotor Type** you select.

To define the rotor pole:

1. To open the **Rotor Pole Data Properties** window, double-click the **Machine-Rotor-Pole** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. For all pole types except type **4**, enter the ratio of the actual arc distance in relation to the maximum possible arc distance in the **Embrace** field. This value is between 0 and 1.



3. For pole type **4**, enter the shaft diameter of the rotor in the **Shaft Diameter** field.
4. For pole types **1**, **2**, and **3**, enter the distance from the center of the rotor to the polar arc center in the **Offset** field. Enter 0 for a uniform air gap.



5. For pole type **5**, enter the thickness of the bridge across the two poles in the **Bridge** field.
6. For pole type **5**, enter the width of the rib supporting the bridge in the **Rib** field.
7. Select the type of magnet to use in the rotor pole from the **Magnet Type** pull-down menu.
8. For pole types **4** and **5**, enter the width of the magnet in the **Magnet Width** field.
9. Enter the maximum radial thickness of the magnet in the **Magnet Thickness** field.
10. Click **OK** to close the **Properties** window.

Rotor Pole Data for Adjust-Speed Synchronous Machines

To access the pole rotor data, double-click the **Machine-Rotor-Pole** entry in the project tree.

The **Rotor Pole Data Properties** window contains the following fields:

Embrace	The pole embrace. For pole types **1**, **2**, **3**, and **5**.
Shaft Diameter	The shaft diameter of the rotor. For pole type **4**.
Offset	The pole-arc center offset from the rotor center (0 for a uniform air gap). For pole types **1**, **2**, and **3**.
Bridge	The thickness of the bridge across two adjacent poles. For pole type **5**.
Rib	The width of the rib at the center of two adjacent poles that support the bridge. For pole type **5**.
Magnet Type	The type of magnet. Click the button to open the **Select Definition** window. For all pole types.
Magnet Width	The maximum width of the magnet. For pole types **4** and **5**.
Magnet Thickness	The maximum thickness of the magnet. For all pole types.

Defining the Shaft Data for an Adjust-Speed Synchronous Machine

To define the shaft:

1. To open the **Shaft Data Properties** window, double-click the **Machine>Shaft** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select or clear the **Magnetic Shaft** check box to specify whether or not the shaft is to be made of magnetic material.
3. Click **OK** to close the **Properties** window.

Shaft Data for Adjust-Speed Synchronous Machines

To access the shaft data, double-click the **Machine>Shaft** entry in the project tree.

The **Shaft Data Properties** window contains the following fields:

Magnetic Shaft	Select or clear this check box to indicate whether or not the shaft is made of magnetic material.

Setting Up Analysis Parameters for an Adjust-Speed Synchronous Machine

To define the solution data:

1. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to open the Solution Setup window.
2. Click the **General** tab. The **Operation Type** is automatically set to **Motor** for this machine type.

Note	To enable selection of **Generator** for Adjust-Speed Synchronous Machines, the machine **Control Type** must be set to **AC** in its Properties window.

   a. If Motor was selected for the Operation Type, select the **Load Type** used in the motor from the following options:

Const Speed	The speed remains constant in the motor.
Const Power	The output power remains constant in the motor.
Const Torque	The torque remains constant regardless of the speed. In this case, $T_{load} = T_{rated}$, given by the output power divided by the given rated speed.
Linear Torque	The torque increases linearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})$ where T_{rated} is given by the output power divided by the given rated speed.
Fan Load	The load varies nonlinearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})^2$ where T_{rated} is given by the output power divided by the given rated speed.

   b. If Generator was selected for the Operation Type, select the **Load Type** used in the generator from the following options: Infinite Bus or Independent Generator.

Infinite Bus	When the generator is connected to an infinite bus of the power system, the terminal voltages remain constant and the output power is determined by the output currents.
Independent Generator	The generator supplies electrical power to a load inductor. For an AC generator, the resistance and inductance of the load inductor are determined by the rated output power, the rated voltage, and the rated power factor. For a DC generator, the resistance of the load resistor is determined by the rated output power and the rated voltage.

3. Enter the output power developed at the shaft of the motor in the **Rated Output Power** field.
4. Enter either the RMS line-to-line voltage (for AC control type), or the DC voltage (for DC and PWM control types) in the **Rated Voltage** field.
5. Enter the desired output speed of the motor at the load point in the **Rated Speed** field.

6. Enter the temperature at which the system functions in the **Operating Temperature** field.
7. Click the **Adjust-Speed Synchronous Machine** tab and select either **Time** or **Frequency** as the **Domain** for the solution.
8. Click **OK** to close the **Solution Setup** window.

Related Topics

Solution Data for Adjust-Speed Synchronous Machines

Solution Data for Adjust-Speed Synchronous Machines

To access the solution data, right-click **Analysis** in the project tree, and click **Add Solution Setup**. For this machine type, there is only one tab, the **General** tab.

The **Solution Setup** window contains the following fields:

Operation Type	The operation type is automatically set to **Motor** for this machine type.
Load Type	Select from **Const Speed**, **Const Power**, **Const Torque**, **Linear Torque**, and **Fan Load**. The default is **Const Power**.
Rated Output Power	Type a value for the rated output voltage, and select the units.
Rated Voltage	Type a value for the rated voltage, and select the units.
Rated Speed	Type a value for the rated speed, and select the units.
Operating Temperature	Type a value for the operating temperature, and select the units.

Related Topics

Setting Up Analysis Parameters for an Adjust-Speed Synchronous Machine

Permanent-Magnet DC Motors

After you have selected **Permanent-Magnet DC Motors** as your model type, you need to define the following:

- General data, such as the voltage, speed, and circuit type of the model.
- Stator data, such as the diameter, slot dimensions, and skew width of the stator.
- Stator pole data, such as its associated pole dimensions, type of steel, and pole magnet specifications.
- Rotor data, such as the slot types and dimensions, slot data, and windings.

- Commutator and brush data, such as the commutator dimensions and brush length.
- Shaft data
- Solution data, such as rated output voltage and frequency.

By option, you can:

- Add a machine housing.

Analysis Approach for PMDC Motors

For a permanent-magnet DC motor, the stator is equipped with P pairs of permanent magnets, creating P pairs of alternating north and south poles. The distribution of the magnetic field produced by the permanent magnet's field flux is fixed with respect to the stator. The rotor is equipped with a distributed winding connected to a commutator that revolves together with the rotor.

A system of brushes is kept in permanent electrical contact with the commutator. When DC current is applied to the rotor winding (via the brushes and commutator), a torque is produced by the interaction of the rotor (armature) currents and the field produced by the permanent magnets.

The commutator causes the armature to create a magnetic flux distribution that is fixed in space and whose axis is perpendicular to the axis of the field flux produced by the permanent magnets. For these motors, the commutator acts as a mechanical rectifier.

The performance of a permanent-magnet DC (PMDC) motor is computed by DC analysis only. The voltage equation of a PMDC motor is:

$$U = Ub + R1 * I + E$$

where Ub is the voltage drop of one-pair brushes, R1 is the armature resistance, $E = Ke * \omega$ is the back emf with *Ke* the back-emf constant in Vs/rad, and ω is the speed in rad/s. For a given speed ω, armature current can be computed based on the applied voltage *U*, as shown below:

$$I = (U - Ub - Ke * \omega)/R1$$

The shaft torque T2 is computed by:

$$T2 = Kt * I - Tfw$$

where *Kt* is the torque constant in Nm/A, which is numerically the same as *Ke*, and *Tfw* is the frictional torque.

The output power (mechanical power) is:

$$P2 = T2 * \omega$$

The input power (electrical power) is:

$$P1 = P2 + Pfw + Pcua + Pb + PFe$$

where *Pfw*, *Pcua*, *Pb*, and *PFe* are frictional and wind loss, armature copper loss, brush drop loss, and iron-core loss, respectively.

The efficiency is:

$$eff = P2/P1 * 100\%$$

Defining a Permanent-Magnet DC Motor

The general procedure for defining a permanent-magnet DC motor is as follows:

1. Insert the permanent-magnet DC motor into a new or existing project.
2. Double-click the **Machine** entry in the project tree to define the general data.
3. Double-click the **Machine>Stator** entry in the project tree to define the stator geometry.
4. Double-click the **Machine-Stator-Pole** entry in the project tree to define the stator pole dimensions.
5. Double-click the **Machine>Rotor** entry in the project tree to define the rotor geometry.
6. Double-click the **Machine-Rotor-Slot** entry in the project tree to define the rotor slot dimensions.
7. Double-click the **Machine-Rotor-Winding** entry in the project tree to define the rotor windings and conductors.
8. Double-click the **Machine>Commutator** entry in the project tree to define the commutator and brush data.
9. Double-click the **Machine>Shaft** entry in the project tree to define the magnetism of the shaft.
10. Optionally, for Inner Rotor machines only, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
11. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
12. Choose **File>Save** to save the project.
13. Choose **RMxprt>Analyze** to analyze the design.

Note	When you place the cursor over an entry field in the data windows, a brief description of that field appears in the status bar at the bottom of the **RMxprt** window.

Once analyzed, the model can be viewed in the Maxwell 2D Modeler, or it can be used to create a new Maxwell 2D project, and a new Maxwell 3D project.

Refer to the *Permanent-Magnet DC Motor Problem* application note, on the technical support page of the Ansys web site, for a specific example of a permanent-magnet DC motor problem.

Defining the General Data for PMDC Motors

Use the **General** window to specify the rated output power, voltage values, circuit type, and speed of the DC motor.

To define the general data:

1. To open the **General Data Properties** window, double-click the **Machine** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the number of poles for the machine in the **Number of Poles** field. This value is the total number of poles in the stator (or the number of pole pairs multiplied by two).
3. Enter the energy loss due to friction at the given speed in the **Frictional Loss** field.

Note	To use the **Brush Press** and **Frictional Coefficient** fields when you define the commutator and brush later in the **Commutator/Brush Data** window, enter **0** here for the **Friction Loss**.

4. Enter the wind loss due to air resistance measured at the reference speed in the **Wind Loss** field.
5. Enter the given speed in the **Reference Speed** field.
6. Click **OK** to close the **Properties** window.

General Data for PMDC Motors

To access the general data, double-click the **Machine** entry in the project tree.

The **General Data Properties** window for a three-phase induction motor contains the following fields:

Machine Type	The machine type you selected when inserting a new RMxprt design (DC Permanent Magnet Motor).
Number of Poles	The number of poles the machine contains.
Rotor Position	Select whether the rotor is an **Inner Rotor** or **Outer Rotor**.
Frictional Loss	The frictional energy loss (due to friction) measured at the reference speed.
Wind Loss	The wind loss (due to air resistance) measured at the reference speed.
Reference Speed	The given speed of reference.

Defining the Stator Data for a PMDC Motor

The stator is the outer lamination stack where the polyphase voltage windings reside.

To define the general stator data:

1. To open the **Stator Data Properties** window, double-click the **Machine>Stator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the **Outer Diameter** of the stator.
3. Enter the **Inner Diameter** of the stator.
4. Enter the length of the stator core in the **Length** field.
5. Enter the stacking factor for the stator core in the **Stacking Factor** field. This value is a ratio of the effective magnetic length of the core, and ranges from 0 to 1. It is defined as the total length minus the total insulation from the laminations, divided by the total length. A value of 1 indicates that the stator is not laminated.
6. Click **OK** to close the **Properties** window.

Stator Data for PMDC Motors

To access the general stator data, double-click the **Machine>Stator** entry in the project tree.

The **Stator Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the stator core.
Inner Diameter	The inner diameter of the stator core.
Length	The length of the stator core.
Stacking Factor	The stacking factor of the stator core.

Defining the Stator Pole for a PMDC Motor

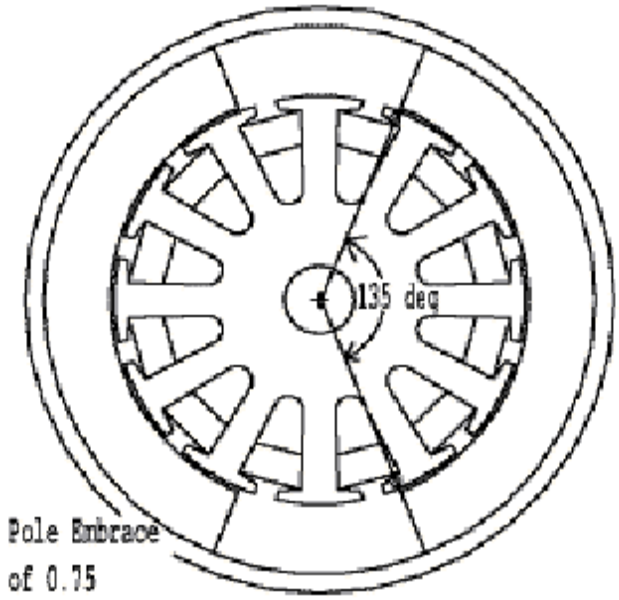
The rotor pole drives the electromagnetic field which is coupled with the stator windings. Use the **s Stator Pole Data Properties** window to define the stator pole.

To define the rotor pole:

1. To open the **Stator Pole Data Properties** window, double-click the **Machine-Stator-Pole** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)

Note	For a two-pole machine, a pole embrace of 0.75 yields a magnet with a span of 135 degrees (based on 0.75*180 degrees).

2. Enter the ratio of the actual arc distance in relation to the maximum possible arc distance in the **Embrace** field. This value is between 0 and 1.



3. Enter the distance from the center of the stator to the magnet arc center in the **Offset** field. Enter 0 for a uniform air gap.
4. To select the type of magnet to use in the rotor pole:
   a. Click the **Magnet Type** button.
      The **Select Definition** window appears.
   b. Select or define a material for the magnet type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
5. Enter the length of the magnet in the axial direction in the **Magnet Length** field.
6. Enter the maximum radial thickness of the magnet at the center of the pole in the **Magnet Thickness** field. To control the flux, the magnet's thickness may vary.
7. Click **OK** to close the **Properties** window.

Stator Pole Data for PMDC Motors

To access the stator pole data, double-click the **Machine-Stator-Pole** entry in the project tree.

The **Stator Pole Data Properties** window contains the following fields:

Embrace	The pole embrace.
Offset	The pole-arc center offset from the stator center (0 for a uniform air gap).
Magnet Type	The type of magnet. Click the button to open the **Select Definition** window.
Magnet Length	The maximum length of the magnet.
Magnet Thickness	The maximum thickness of the magnet.

Defining the Rotor Data for a PMDC Motor

The rotor is equipped with slots containing copper conductors that are connected to the commutator. The commutator acts as a mechanical rectifier in the motor.

Use the **Rotor Data Properties**, **Rotor Slot Data Properties**, and **Rotor Winding Data Properties** windows to define the rotor slots, windings, and dimensions.

To define general rotor data:

1. To open the **Rotor Data Properties** window, double-click the **Machine>Rotor** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the stacking factor for the rotor core in the **Stacking Factor** field.
3. Enter the **Number of Slots** in the rotor.
4. Select the **Slot Type**:
   a. Click the button for the **Slot Type**.
   The **Select Slot Type** window appears.
   b. Select a slot type (available types include 1 through 4).

Note	When you place the mouse cursor over the slot type, a schematic outline of the slot appears.

   c. Click **OK** to close the **Select Slot Type** window and return to the **Properties** window.
5. Enter the outer diameter of the rotor in the **Outer Diameter** field.
6. Enter the inner diameter of the rotor in the **Inner Diameter** field.
7. Enter the length of the rotor core in the **Length** field.
8. Select a **Steel Type** for the rotor core:
   a. Click the button for **Steel Type**.
   The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
9. Enter the skew width, measured in slot number, in the **Skew Width** field.
10. Click **OK** to close the **Properties** window.

Rotor Data for PMDC Motors

To access the general rotor data, double-click the **Machine>Rotor** entry in the project tree.

The **Rotor Data Properties** window contains the following fields:

Stacking Factor	The stacking factor of the rotor core.
Number of Slots	The number of slots the rotor core contains.
Slot Type	The type of slots in the rotor core. Click the button to open the **Select Slot Type** window.

Outer Diameter	The outer diameter of the rotor core.
Inner Diameter	The inner diameter of the rotor core.
Length	The length of the rotor core.
Steel Type	The steel type of the rotor core. Click the button to open the **Select Definition** window.
Skew Width	The skew width measured in slot number.

Defining the Rotor Slots for a PMDC Motor

To define the physical dimensions of the rotor slots:

1. To open the **Rotor Slot Data Properties** window, double-click the **Machine-Rotor-Slot** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Optionally, to automatically design the dimensions of slots **Hs2**, **Bs1**, and **Bs2**, select the **Auto Design** check box.
3. Enter the available slot dimensions.

Hs0	Always available.
Hs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.
Bs0	Always available.
Bs1	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Bs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Rs	Rs is added when the slot type is 3 or 4.

4. Click **OK** to close the **Properties** window.

Rotor Slot Data for PMDC Motors

To access the stator slot data, double-click the **Machine-Rotor-Slot** entry in the project tree.

The **Rotor Slot Data Properties** window contains the following fields:

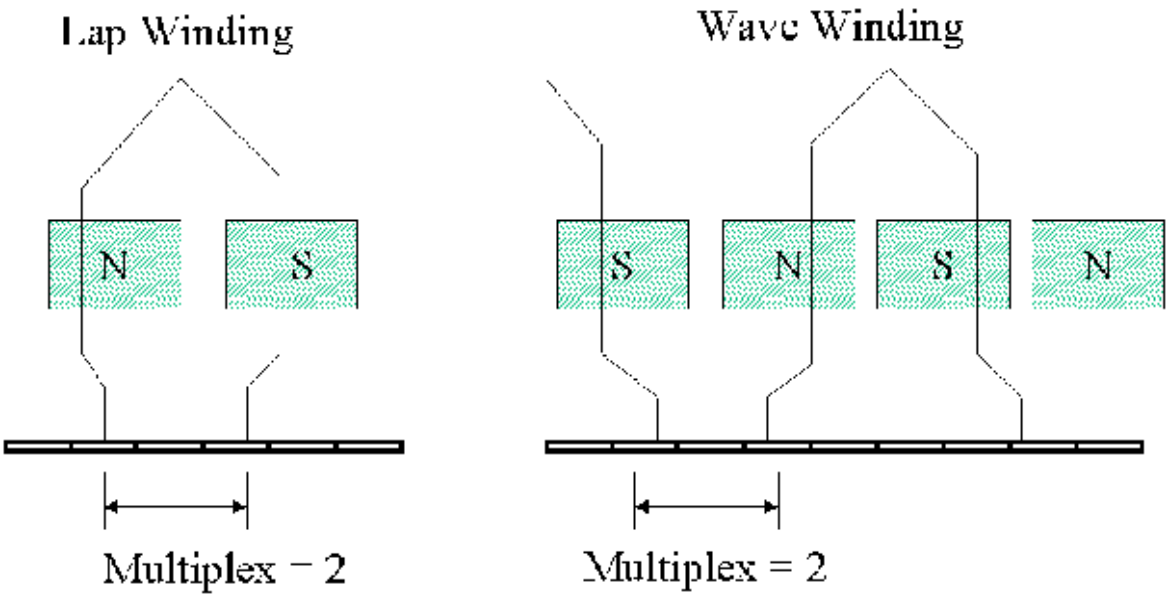
Auto Design	Select or clear this to enable or disable auto-design of slots **Hs2**, **Bs1**, and **Bs2**. When this check box is selected, only two other fields appear in the window: **Hs0** and **Bs0**.
Hs0	A slot dimension (see the diagram shown in the modeling window when **Machine-**

	Rotor-Slot is selected).
Hs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Bs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Bs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Bs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Rs	A slot dimension. (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected). Rs is added when the slot type is 3 or 4.
Rs	A slot dimension. (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected). Rs is added when the slot type is 3 or 4.

Defining the Rotor Windings and Conductors for a PMDC Motor

To define the rotor windings, wires, and conductors:

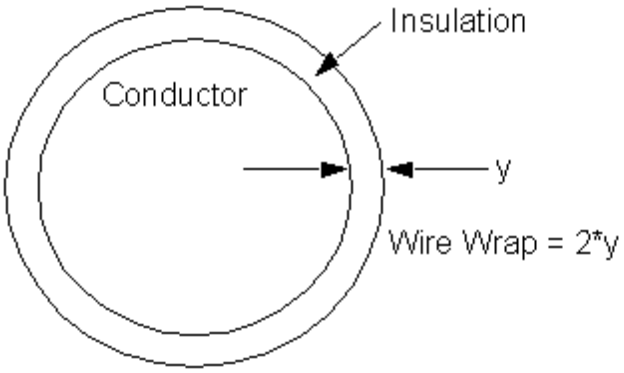
1. To open the **Rotor Slot Winding Properties** window, double-click the **Machine-Rotor-Winding** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Click the **Winding** tab.
3. Select a **Winding Type**:
   a. Click the button for **Winding Type**.
   The **Winding Type** window appears.
   b. Select from one of the following three types of winding:
      - **Lap**
      - **Wave**
      - **Frog Leg**
4. Enter the number of windings in the **Multiplex Number** field (1 for a single winding, 2 for double windings, 3 for triple windings). For a lap winding, the multiplex number is the number of commutators between the start and end of one winding, and the number of parallel branches is equal to the number of poles multiplied by the multiplex number. For a wave winding, the number of parallel branches equals the multiplex number multiplied by two.



5. Enter the number of virtual slots per each real slot in the **Virtual Slots** field. The rotor is assumed to have two layers of conductors, an upper and a lower layer. Each layer of conductors can have a number of windings, which are referred to as virtual slots.

Note	For example, the upper and lower layer can have two windings each, which would yield a virtual slot number of two; for a 12 slot machine, this would yield 24 commutation segments.

6. Enter the total number of conductors in each rotor slot in the **Conductors per Slot** field. This value is the number of turns per coil multiplied by the number of layers. This value is the total number of conductors in one real full rotor slot.
7. Enter the coil pitch, measured in number of slots, in the **Coil Pitch** field. The coil pitch is the number of slots separating one winding. For example, if a coil starts in slot 1 and ends in slot 6, it has a coil pitch of 5.
8. Enter the number of wires per conductor in the **Number of Strands** field. Enter 0 to have RMxprt auto-design this value.
9. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



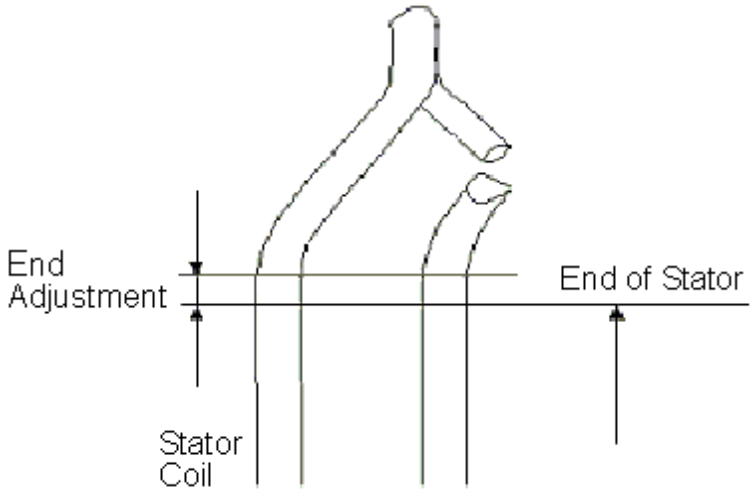
10. Select the **Wire Size**:
    a. Click the button for **Wire Size**.
    The **Wire Size** window appears.
    b. Select a value from the **Wire Diameter** pull-down list.
    c. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

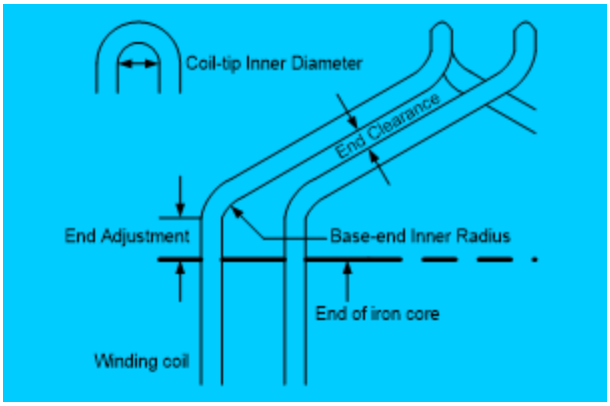
<*number*>	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.
USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.
AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.
MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21mm and 2 with a diameter of 0.13mm.

The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>General Options>Machines**command. When you are done setting the wire size, click **OK** to close the **Wire Size** window and return to the **Properties** window.

11. Click the **End/Insulation** tab.
12. Select or clear the **Input Half-turn Length** check box.
13. Do one of the following:
    - If you selected **Input Half-turn Length**, then enter the half-turn length of the armature winding in the Half Turn Length field.

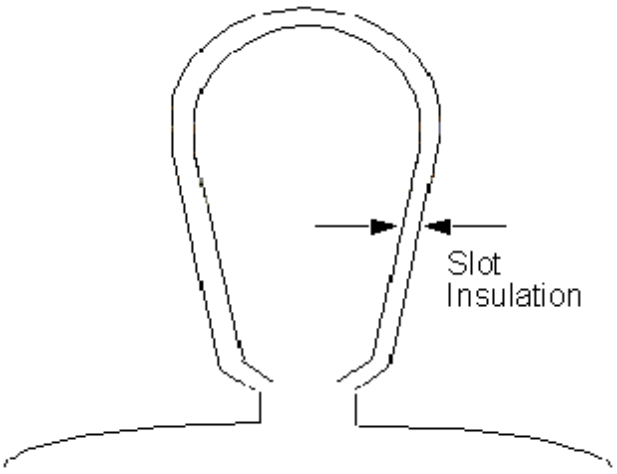
- If you cleared **Input Half-turn Length**, then enter the end length adjustment of the stator coils in the **End Adjustment** field. The end adjustment is the distance one end of the conductor extends vertically beyond the end of the stator.





14. Enter the inner radius of the base corner in the **Base Inner Radius** field.
15. Enter the inner diameter of the coil tip in the **Tip Inner Diameter** field.
16. Enter the distance between two rotor coils in the **End Clearance** field.

17. Enter the thickness of the slot liner insulation in the **Slot Liner** field.



18. Enter the thickness of the wedge insulation in the **Wedge Thickness** field.
19. Enter the thickness of the insulation layer in the **Layer Insulation** field.
20. Enter the limited slot fill factor for the wire design in the **Limited Fill Factor** field. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.
21. Select the type of equalizer connection from the **Equalizer Connection** pull-down menu. Select from **None**, **Half**, or **Full**.
22. Click **OK** to close the **Properties** window.

Defining Different Size Wires for a PMDC Motor

Use the **Gauge** option if you have a conductor that is made up different size wires.

To define different size wires:

1. In the **Wire Size** window, select **MIXED** from the **Gauge** pull-down menu.
2. Select either **Round** or **Rectangular** as the **Wire Type**.
3. Enter the appropriate wire data in the table:
   - For a round wire:
     - Enter the **Diameter** in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this diameter.
   - For a rectangular wire:
     - Enter the **Width** of the wire in the table.
     - Enter the **Thickness** of the wire in the table.
     - Enter the **Fillet** value in the table.

- Enter a **Number** in the table to specify how many of the conductor's wires have this data.

4. Click **Add** to add the new wire data.
5. Repeat steps 3 and 4 for each size wire you want to add.
6. When you are finished defining the wires, click **OK** to close the **Wire Size** window and return to the RMxprt **Properties** window.

Note	For example, if one conductor is made up of 5 wires, and 3 of those wires have a diameter of 0.21mm, and the other 2 have a diameter of 0.13mm, then the mixed wire size table will have two lines. The first line will list Diameter = 0.21 and Number = 3. The second line will list Diameter = 0.13 and Number = 2. An equivalent wire diameter is displayed as Wire Size value in the Winding tab in the Properties window.

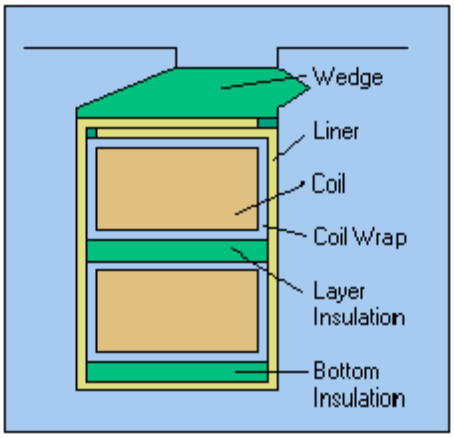
Rotor Winding Data for PMDC Motors

To access the stator winding data, double-click the **Machine-Rotor-Winding** entry in the project tree.

The **Rotor Winding Data Properties** window contains the following fields:

Winding tab	**Winding Type**	The type of rotor winding. Click the button to open the **Winding Type** window and choose from **Lap**, **Wave**, and **Frog Leg**.
	Multiplex Number	Single, double, or triple windings (1, 2, or 3).
	Virtual Slots	The number of virtual slots per real slot.
	Conductors per Slot	The number of conductors per rotor slot (0 for auto-design).
	Coil Pitch	The coil pitch measured in number of slots.
	Number of Strands	The number of wires per conductor (0 for auto-design).
	Wire Wrap	The thickness of the double-sided wire wrap (0 to automatically obtain this value from the wire library).
	Wire Size	The diameter of the wire (0 for auto-design). Click the button to open the **Wire Size** window where you can specify units, wire type, diameter, and gauge.

End/Insulation tab	**Input Half-turn Length**	Select or clear this check box to specify whether or not you want to enter the half-turn length. When this check box is selected, the **Half Turn Length** field appears the next time you open the **Properties** window. When this check box is selected, the **End Adjustment** field appears instead.
	Half Turn Length	The half-turn length of the armature winding.
	Base Inner Radius	The inner radius of the base corner.
	Tip Inner Diameter	The inner diameter of the coil tip.
	End Clearance	The end clearance between two adjacent coils.
	Slot Liner	The thickness of the slot liner insulation.
	Wedge Thickness	The thickness of the wedge insulation.
	Layer Insulation	The thickness of the insulation layer.
	Limited Fill Factor	The limited slot fill factor for the wire design. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.
	Equalizer Connection	The connection type of the equalizer. Select from **None**, **Half**, or **Full**.



Defining the Commutator and Brush for a PMDC Motor

The commutator allows current transfer between DC terminals or brushes and the rotor coils, providing the current to the system as a function of rotation. Due to the action of the commutator, the corresponding magnetic field has a fixed distribution with respect to the stator.

To define the commutator and brush pairs:

1. To open the **Commutator Data Properties** window, double-click the **Machine>Commutator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Click the **Commutator** tab.
3. Select **Cylinder** or **Pancake Type** as the **Commutator Type**.

Note	When you place the mouse cursor over the commutator type, an outline of the commutator appears.

4. For **Cylinder** commutators, do the following:
   a. Enter the **Commutator Diameter**.
   b. Enter the **Commutator Length**.
5. For **Pancake** commutators, do the following:
   a. Enter the **Outer Diameter**.
   b. Enter the **Inner Diameter**.
6. Enter the thickness of the insulation between two consecutive commutator segments in the **Commutator Insulation** field.
7. Click the **Brush** tab.
8. Enter the **Brush Width**.
9. Enter the **Brush Length**.
10. Enter the number of brush pairs when using a wave armature winding in the **Brush Pairs** field.
11. Enter the angle of displacement from the neutral axis, in mechanical degrees, in the **Brush Displacement** field.

Note	The brush displacement is positive for the counter-clockwise direction. For example, if the rotor turns clockwise and the brush displacement is also clockwise, then the angle is negative; if the rotor turns clockwise but the brush displacement is counter-clockwise, then the angle is positive.

12. Enter the voltage drop across one brush pair in the **Brush Drop** field.
13. Enter the mechanical pressure of the brushes as they press against the commutator in the **Brush Press** field.

14. Enter the **Frictional Coefficient** of the brush.

Note	If the **Friction Loss** field is used in the **General** window, the **Brush Press** and **Frictional Coefficient** fields will be hidden in the **Commutator/Brush** window. These fields are shown only when the **Friction Loss** field in the **General** window is set to zero.

15. Click **OK** to close the **Properties** window.

Commutator and Brush Data for PMDC Motors

To access the commutator and brush data, double-click the **Machine>Commutator** entry in the project tree.

The **Commutator Data Properties** window contains the following fields:

Commutator tab	**Commutator Type**	The type of commutator. Click the button to open the **Select Commutator Type** window and select from **Cylinder** or **Pancake**.
	Commutator Diameter	For a **Cylinder** commutator type, the diameter of the commutator.
	Commutator Length	For a **Cylinder** commutator type, the length of the commutator.
	Outer Diameter	For a **Pancake** commutator type, the outer diameter of the commutator.
	Inner Diameter	For a **Pancake** commutator type, the inner diameter of the commutator.
	Commutator Insulation	The thickness of the insulation between the two commutator bars.
Brush tab	**Brush Width**	The width of the brush.
	Brush Length	The length of the brush.
	Brush Pairs	The number of brush pairs.
	Brush Displacement	The displacement of the brush from the neutral position, in mechanical degrees (positive for anti-rotating direction).
	Brush Drop	The voltage drop across a one-pair brush.
	Brush Press	The brush press per unit area. (Available only when **Frictional Loss** is set to zero for the machine.)
	Frictional Coefficient	The frictional coefficient of the brush. (Available only when **Frictional Loss** is set to zero for the machine.)

Defining the Shaft Data for a PMDC Motor

To define the shaft:

1. To open the **Shaft Data Properties** window, double-click the **Machine>Shaft** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select or clear the **Magnetic Shaft** check box to specify whether or not the shaft is to be made of magnetic material.
3. Click **OK** to close the **Properties** window.

Shaft Data for PMDC Motors

To access the shaft data, double-click the **Machine>Shaft** entry in the project tree.

The **Shaft Data Properties** window contains the following fields:

Magnetic Shaft	Select or clear this check box to indicate whether or not the shaft is made of magnetic material.

Setting Up Analysis Parameters for a PMDC Motor

To define the solution data:

1. To open the **Solution Setup** window, right-click **Analysis** in the project tree, and click **Add Solution Setup**.
2. Click the **General** tab. The **Operation Type** is automatically set to **Motor** for this machine type.
3. Select the **Load Type** used in the motor from the following options:

Const Speed	The speed remains constant in the motor.
Const Power	The output power remains constant in the motor.
Const Torque	The torque remains constant regardless of the speed. In this case, $T_{load} = T_{rated}$, given by the output power divided by the given rated speed.
Linear Torque	The torque increases linearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})$ where T_{rated} is given by the output power divided by the given rated speed.
Fan Load	The load varies nonlinearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})^2$ where T_{rated} is given by the output power divided by the given rated speed.

4. Enter the output power developed at the shaft of the motor in the **Rated Output Power** field.
5. Enter the RMS line-to-line voltage in the **Rated Voltage** field.
6. Enter the desired output speed of the motor at the load point in the **Rated Speed** field.

7. Enter the temperature at which the system functions in the **Operating Temperature** field.
8. Click **OK** to close the **Solution Setup** window.

Related Topics

Solution Data for PMDC Motors

Solution Data for PMDC Motors

To access the solution data, right-click **Analysis** in the project tree, and click **Add Solution Setup**. For this machine type, there is only one tab, the **General** tab.

The **Solution Setup** window contains the following fields:

Operation Type	The operation type is automatically set to **Motor** for this machine type.
Load Type	Select from **Const Speed**, **Const Power**, **Const Torque**, **Linear Torque**, and **Fan Load**. The default is **Const Power**.
Rated Output Power	Type a value for the rated output voltage, and select the units.
Rated Voltage	Type a value for the rated voltage, and select the units.
Rated Speed	Type a value for the rated speed, and select the units.
Operating Temperature	Type a value for the operating temperature, and select the units.

Related Topics

Setting Up Analysis Parameters for a PMDC Motor

Three-Phase Synchronous Machines

After you have selected **Three-Phase Synchronous Machines** as your model type, you need to define the following:

- General data, such as the unit system, power, and voltage.
- Stator data, such as the slot types and dimensions, stator diameter, skew width, and laminations.
- Optional stator Vent data.
- Winding data, such as the parallel branches, conductors, and wire dimensions.
- Rotor pole data, such as its associated pole-body dimensions and air gaps.
- Optional Rotor damper data, such as the damper dimensions, rings, and material properties.
- Rotor winding data and the winding control parameters.
- Shaft Data

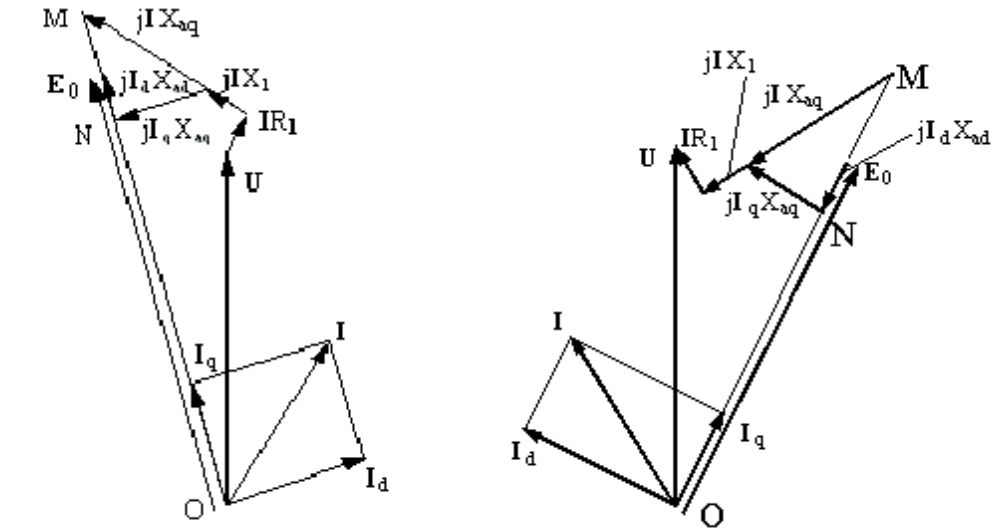
- Optional machine housing.
- Solution data, such as specifying motor or generator application, and rated output voltage and frequency.

Also see the Analysis Approach for Three-Phase Synchronous Machines.

Analysis Approach for Three-Phase Synchronous Machines

The three-phase salient-pole synchronous electric machine has two types: the generator and the motor. Their basic structures are the same. Three-phase synchronous generators are the main source of electrical energy for industrial, commercial, and private use. They receive mechanical energy at the shaft and transform it into electrical energy. The rotor is equipped with a multi-pole winding excited by a DC source. The stator is equipped with a three-phase winding that has a sinusoidal spatial distribution. The spinning rotor produces a rotating magnetic field in the air gap of the machine. The frequency of the voltage induced in the stator is given by $f=pv$, where p is the number of pairs of poles, and v is the velocity of the rotor. The machine is capable of producing both active and reactive power as required by the load connected at the stator phasor.

The three-phase salient-pole synchronous electric machine has two types: the generator and the motor. Their basic structures are the same. Usually the frequency-domain phasor diagram is adopted to analyze the characteristics. The phasor diagram for a generator is shown on the left and that for a motor is shown on the right.



In the figure, R1, X1, Xad, and Xaq are armature resistance, armature leakage reactance, d-axis armature reactance, and q-axis armature reactance, respectively. Xad is nonlinear, while a linearized value is used in the phasor diagram. Taking the input voltage **U** as the reference phasor, for a given current:

$$\mathbf{I} = I\angle -\varphi$$

where φ is the power factor angle, a phasor represented by OM can be derived by:

U + **I**(R1 + jX1 + jXaq)

The direction of $\mathbf{E}_0$ can, therefore, be obtained. Taking the power angle, the angle that **U** legs $\mathbf{E}_0$, as θ, then the angle that **I** legs $\mathbf{E}_0$ is:

$$\psi = \theta + \varphi$$

The d- and q-axis currents are then represented by the following:

$$Id = I * \sin(\psi)$$

$$Iq = I * \cos(\psi)$$

The phasor length ON represents the d-axis back EMF from d-axis resultant flux linkage and is used to determine the d-axis field saturation. Then a frozen method is applied to derive $\mathbf{E}_0$, X_{ad}, and exciting current I_f.

The output power (electric power) is directly computed from voltage and current as:

$$P2 = 3*U*I*\cos(\varphi)$$

The input power (mechanical power) is defined as:

$$P1 = P2 + Pfw + Pcua + PFe + Padd + Pcuf + Pex$$

where Pfw, Pcua, PFe, Padd, Pcuf and Pex are frictional and wind loss, armature copper loss, iron-core loss, additional loss, field winding copper loss, and exciter loss, respectively.

The input mechanical shaft torque is:

$$T1 = P1/\omega$$

where SYMBOL is synchronous speed in rad/s.

The efficiency is computed by:

```
eff = P2/P1 * 100%
```

Main Features

- Adapted to both Synchronous Motor and Generator

  The structures of the salient-pole synchronous motor and the generator are basically the same, but their phasor relationships and the computation methods are slightly different, their output characteristics data are also different. This is specified in the solution setup.

- Auto Arrangement of Three-phase Windings

  Almost all commonly used three-phase single- and double-layer, half- and whole-type ac windings (including fractional-pitch windings) can be automatically arranged. Users do not need to define coils one by one. RMxprt also supports a double-layer winding with half-turn coils which are auto-arranged in the order of even, odd, even, odd, …, and even, odd, as long as it is physically possible.

  When a designer adopts single-layer whole-coiled windings, RMxprt will perform winding arrangement optimization to minimize the average coil pitch. When asymmetric three-phase windings are used, winding arrangement is optimized in such a way that minimum negative-sequence and zero-sequence components are achieved.

- Winding Editor Supporting Any Single- and Double-Layer Windings

  Besides taking advantage of the winding auto-arrangement function in RMxprt, users can also specify any special winding by using of the Winding Editor function.

  In Winding Editor, through modification of phase belonging, number of turns, in-slot and out-slot number of each coil, it is possible to design single- and double-layer winding arrangement for any purposes.

- Analyze Air-Gap Magnetic Field Distribution

  For both uniform and non-uniform air gaps, Schwarz-Christopher Transformation is adopted to solve for the air-gap magnetic field distribution.

- Analyze EMF Waveform and Total Harmonic Distortion (THD)

  Based on the analysis of the air-gap magnetic field waveform, taking into account coil short pitch, winding distribution, skew slot, winding connection, load effects and other factors, the emf waveforms in the coils and the windings are analyzed to solve for the emf distortion factors.

- Analyze Dynamic Parameters of Damping Winding

  Different from the squirrel-cage winding of the induction machine, the damping winding of the salient-pole synchronous machine is located in the surface of magnetic field poles, which deviates greatly along the d- and the q-axes. Furthermore, the connection of damping bars has several forms. The bars under each pole could be connected, but not connected with those under other poles. All the bars could be connected together. The bars could be connected through end-plate. RMxprt can deal with all those complicated situations and give the dynamic parameters for the damping winding.

Defining a Three-Phase Synchronous Machine

The general procedure for defining a three-phase synchronous machine is as follows:

1. Insert a three-phase synchronous machine into a existing or new project.
2. Double-click the **Machine** entry in the project tree to define the general data.
3. Double-click the **Machine-Stator** entry in the project tree to define the stator geometry.
4. Double-click the **Machine-Stator-Slot** entry in the project tree to define the stator slot dimensions.
5. Optionally, you can add a vent to, or remove an existing vent from the stator. To add a vent, select the stator, and right-click to display the pop-up menu for **Insert Vent**.
6. Double-click the **Machine-Stator-Winding** entry in the project tree to define the stator windings, conductors, and insulation data.
7. Double-click the **Machine-Rotor** entry in the project tree to define the general rotor geometry, the pole data, and the insulation data.
8. Double-click the **Machine-Rotor-Winding** entry in the project tree to define the rotor conductors and windings.
9. Optionally, you can add a damper to the design or remove an existing damper. To add a damper, use **Machine-Insert Damper**. This inserts the damper in the project tree under the rotor. You must then specify the slot type and other properties for the damper.
10. Optionally, you can add a vent to, or remove an existing vent from the rotor. To add a vent select the rotor, and right-click to display the pop-up menu. Use **Insert Vent**.
11. Double-click the **Machine-Shaft** entry in the project tree to define the magnetism of the shaft.
12. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
13. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define this solution data.
14. Choose **File>Save** to save the project.
15. Choose **RMxprt>Analyze** to analyze the design.

Note	When you place the cursor over an entry field, a brief description of that field appears in the status bar at the bottom of the **RMxprt** window.

Once the design is analyzed, the model can be viewed in the Maxwell 2D Modeler, or it can be used to create a new Maxwell 2D project, and a Maxwell 3D design.

Defining the General Data for a Three-Phase Synchronous Machine

Use the **General Data Properties** window to define the power settings, speed, and efficiency of the generator. This window allows you to define the basic parameters of the synchronous generator, such as power, voltage, winding connections, and losses.

To define the general data:

1. To open the **General Data Properties** window, double-click the **Machine** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)

2. Enter the number of poles for the machine in the **Number of Poles** field. This value is the total number of poles in the stator (or the number of pole pairs multiplied by two).
3. Enter the power lost through frictional forces in the **Frictional Loss** field.
4. Enter the wind loss measured at the reference speed in the **Wind Loss** field.
5. Enter the given speed in the **Reference Speed** field.
6. Click **OK** to close the **Properties** window.

General Data for Three-Phase Synchronous Machines

To access the general data, double-click the **Machine** entry in the project tree.

The **General Data Properties** window for a three-phase induction motor contains the following fields:

Machine Type	The machine type you selected when inserting a new RMxprt design (Three Phase Synchronous Machine).
Number of Poles	The number of poles the machine contains.
Frictional Loss	The frictional energy loss (due to friction) measured at the reference speed.
Wind Loss	The wind loss (due to air resistance) measured at the reference speed.
Reference Speed	The given speed of reference.

Defining the Stator for a Three-Phase Synchronous Machine

Use the **Stator** windows to define the slot dimensions, stacking factors, air ducts, and insulation of the stator. The stator is the outer lamination stack where the three-phase windings reside.

To define general stator data:

1. To open the **Stator Data Properties** window, double-click the **Machine>Stator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the **Outer Diameter** of the stator.
3. Enter the **Inner Diameter** of the stator.
4. Enter the length of the stator core in the **Length** field.
5. Enter the stacking factor for the stator core in the **Stacking Factor** field.
6. Select a **Steel Type** for the stator core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.

7. Enter the **Number of Slots** in the stator.
8. Select the**Slot Type**:
   a. Click the button for the **Slot Type**.
      The **Select Slot Type** window appears.
   b. Select a slot type (available types include 1 through 6). Slot types **1** though **4** are filled with round wire. Slot types **5** and **6** are filled with rectangular wire. If **Auto Design** is enabled, the software designs an optimum slot geometry; in this case, you can input the tooth width dimension, and the software determines the slot width accordingly.

Note	When you place the mouse cursor over the slot type, a schematic of the selected type appears, displaying the slot dimension variables.

   c. Click **OK** to close the **Select Slot Type** window and return to the **Properties** window.
9. Enter the number of sectors in the **Lamination Sectors** field.
10. Enter the thickness of the magnetic pressboard in the **Pressboard Thickness** field. Enter **0** for a non-magnetic pressboard.
11. Enter the skew width, measured in slot number, in the **Skew Width** field.
12. Click **OK** to close the **Properties** window.

Stator Data for Three-Phase Synchronous Machines

To access the general stator data, double-click the **Machine>Stator** entry in the project tree.

The **Stator Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the stator core.
Inner Diameter	The inner diameter of the stator core.
Length	The length of the stator core.
Stacking Factor	The stacking factor of the stator core.
Steel Type	The steel type of the stator core. Click the button to open the **Select Definition** window.
Number of Slots	The number of slots the stator core contains.
Slot Type	The type of slots in the stator core. Click the button to open the **Select Slot Type** window.
Lamination Sectors	The number of lamination sectors.
Pressboard Thickness	The magnetic press board thickness (0 for a non-magnetic press board).
Skew Width	The skew width measured in slot number.

Defining Stator Slots for a Three-Phase Synchronous Machine

To define the slot dimensions:

1. To open the **Stator Slot Data Properties** window, double-click the **Machine-Stator-Slot** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Optionally, to automatically design the dimensions of slots **Hs2**, **Bs1**, and **Bs2**, select the **Auto Design** check box.
3. Optionally, to design dimensions of slots **Bs1** and **Bs2** based on the stator tooth width, select the **Parallel Tooth** check box, and enter a value in the **Tooth Width** field.
4. Enter the available slot dimensions.

Hs0	Always available.
Hs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.
Bs0	Always available.
Bs1	Available only when **Auto Design** and **Parallel Tooth** are both cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Bs2	Available only when **Auto Design** and **Parallel Tooth** are both cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.

5. Click **OK** to close the **Properties** window.

Stator Slot Data for Three-Phase Synchronous Machines

To access the stator slot data, double-click the **Machine-Stator-Slot** entry in the project tree.

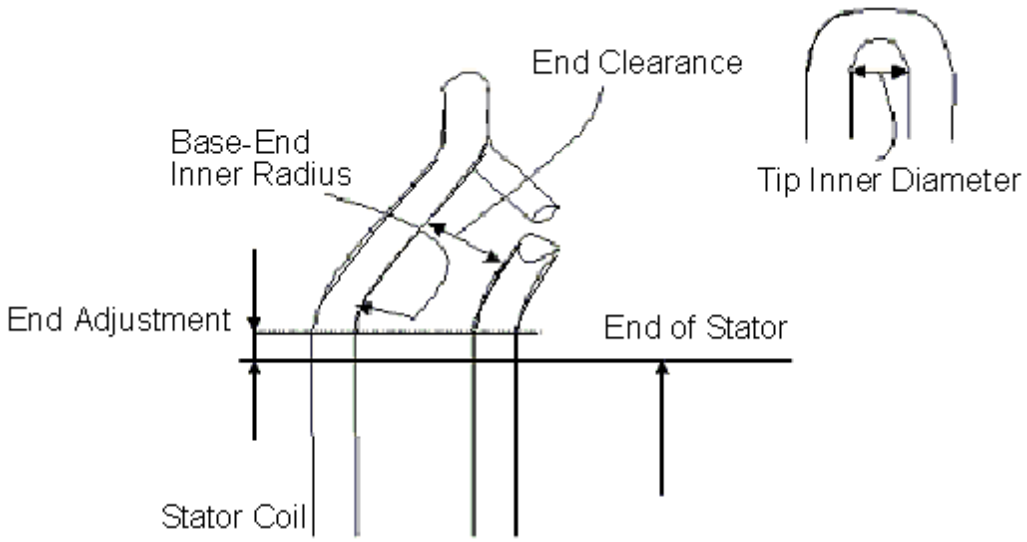
The **Stator Slot Data Properties** window contains the following fields:

Auto Design	Select or clear this to enable or disable auto-design of slots **Hs2**, **Bs1**, and **Bs2**. When this check box is selected, only two other fields appear in the window: **Hs0** and **Bs0**.
Parallel Tooth	Select this to design **Bs1** and **Bs2** based on the tooth width. When this check box is selected, the **Bs1** and **Bs2** fields are removed, and the **Tooth Width** field is added.
Tooth Width	The tooth width for the parallel tooth, on which **Bs1** and **Bs2** are designed.
Hs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Hs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs1	A slot dimension (see the diagram shown in the modeling window when **Machine-**

	Stator-Slot is selected).
Bs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Rs	A slot dimension. (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected). Rs is added when the slot type is 3 or 4.

Defining Stator Windings and Insulation for a Three-Phase Synchronous Machine

Use the **Stator Winding** window to define the stator winding data, such as the coils, wires, insulation, number of parallel branches, and physical dimensions of the windings.

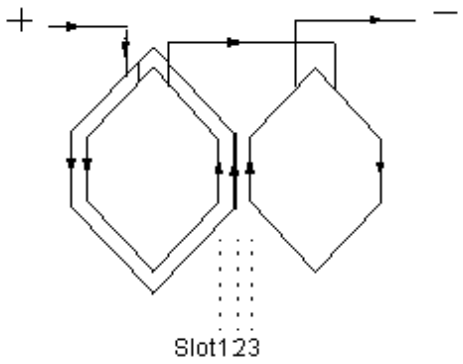
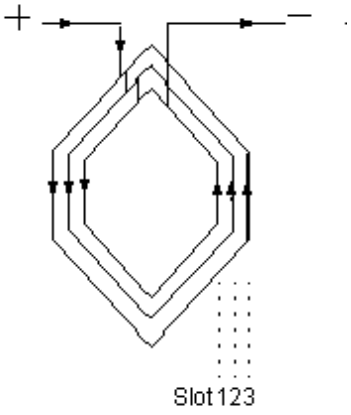


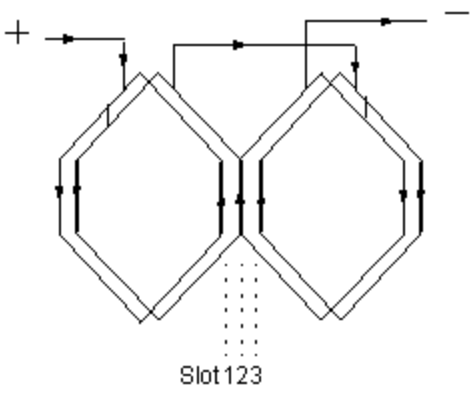
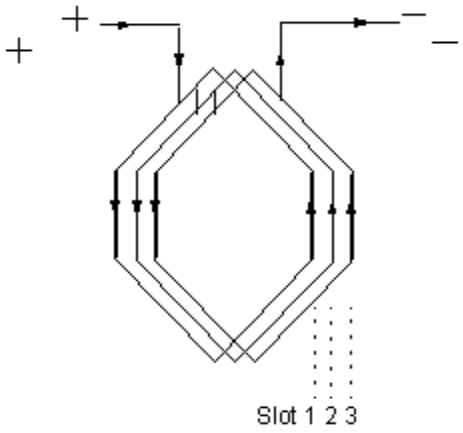
The stator winding data defines the configuration of one phase of the three-phase windings.

To define the stator windings and insulation:

1. To open the **Stator Slot Winding Properties** window, double-click the **Machine-Stator-Winding** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Click the **Winding** tab.
3. Enter the number of layers in the stator winding in the **Winding Layers** field.
4. Select a **Winding Type**:
   a. Click the button for **Winding Type**.
   The **Winding Type** window appears.
   b. Select from one of the following three types of winding:
      - **Whole Coiled**
      - **Half Coiled**
      - **Editor**

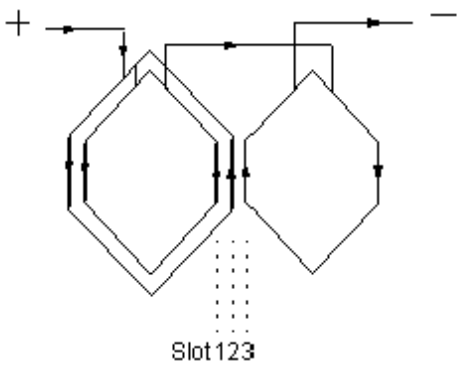
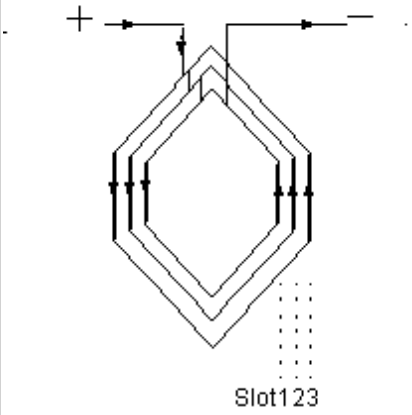
When you place the mouse cursor over a winding button, an outline of the selected winding appears. The following table describes the six types of windings that are possible (three for one-layer and three for two-layer):

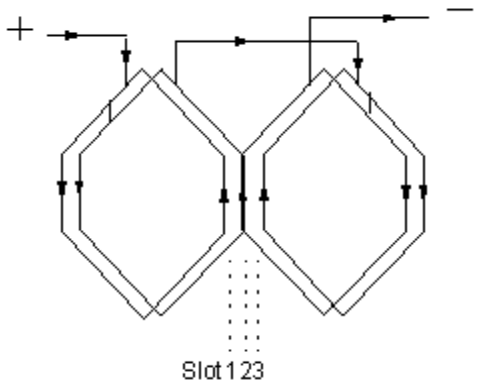
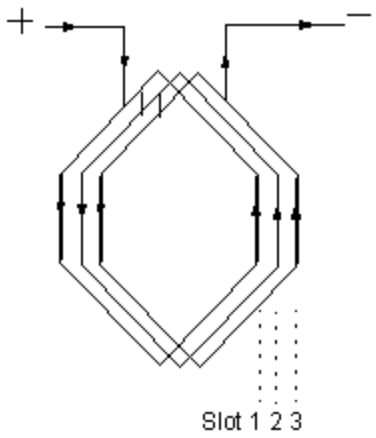
Type	Description
Editor	A user-defined one-layer winding arrangement. You need to set up the winding arrangement for each slot. For this winding type, the following letters are used for the phase windings: • Phase A/A return uses A/X. • Phase B/B return uses B/Y. • Phase C/C return uses C/Z.
Whole Coiled	A one-layer whole-coiled winding: 
Half Coiled	A one-layer concentric half-coiled winding: 
Editor	A user-defined two-layer winding arrangement. When you select **20**, the Winding Editor opens, where you can specify a different winding arrangement for each slot.

Whole Coiled	A two-layer wave winding:  The phase belt for this winding configuration is equal to 360/2*m*, where *m* is the phase number.
Half Coiled	A two-layer half-coiled winding:  There is only one coil per phase per pair of poles.

Note	**Example 1:** A one layer winding arranged in 12 slots should be defined as type **10**, with the following arrangement: **AAZZBBXXCCYY** **Example 2:** A two layer winding arranged in 12 slots should be defined as type **20**, with the following arrangement: **AAZZBBXXCCYY** Only the top layer needs to be defined; the bottom layer will be determined according to the coil pitch.

Once you have clicked a button to select a winding, click **OK** to close the **Winding Type** window and return to the **Properties** window. Select a **Winding Type**. When you place the mouse cursor over a winding, an outline of the selected winding appears. The following winding types are available:

10	A user-defined single-layer winding arrangement. When you select it, enter the winding arrangement, and choose **OK**. For this winding type, the following letters are used for the phase windings: • phase A/A return uses A/X. • phase B/B return uses B/Y. • phase C/C return uses C/Z.
11	A one-layer whole-coiled winding: 
12	A one-layer concentric half-coiled winding: 
20	A user-defined winding arrangement. When you select this type, enter the winding arrangement, and choose **OK**.
21	A two-layer wave winding:

	 The phase belt for this winding configuration is equal to 360/2*m*, where *m* is the phase number.
22	A two-layer winding: 
Note	**Example 1:** A one layer winding arranged in 12 slots should be defined as type **10**, with the following arrangement: **AAZZBBXXCCYY** **Example 2:** A two layer winding arranged in 12 slots should be defined as type **20**, with the following arrangement: **AAZZBBXXCCYY** Only the top layer needs to be defined; the bottom layer will be determined according to the coil pitch.

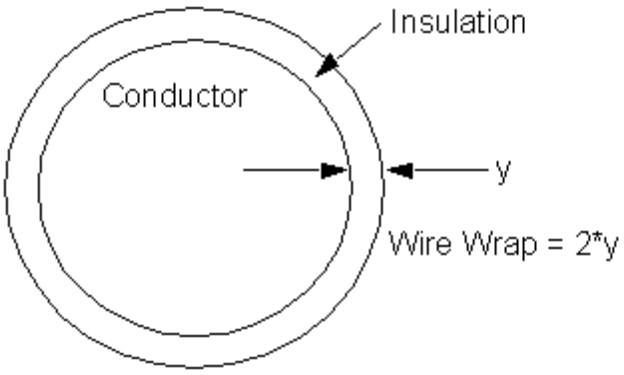
a. Select or enter the number of parallel branches in one phase of the winding in the **Parallel Branches** field.

b. Enter the total number of conductors in each stator slot in the **Conductors per Slot** field. This value is the number of turns per coil multiplied by the number of layers.

c. Enter the coil pitch, measured in number of slots, in the **Coil Pitch** field. The coil pitch is the number of slots separating one winding. For example, if a coil starts in slot 1 and ends in slot 6, it has a coil pitch of 5.

d. Enter the number of wires per conductor in the **Number of Strands** field. Enter 0 to have RMxprt auto-design this value.

e. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



f. Select the **Wire Size**: Click the button for **Wire Size**. The **Wire Size** window appears. Select a value from the **Wire Diameter** pull-down list. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

g.

<*number*>	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.
USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.
AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.
MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21mm and 2 with a diameter of 0.13mm.

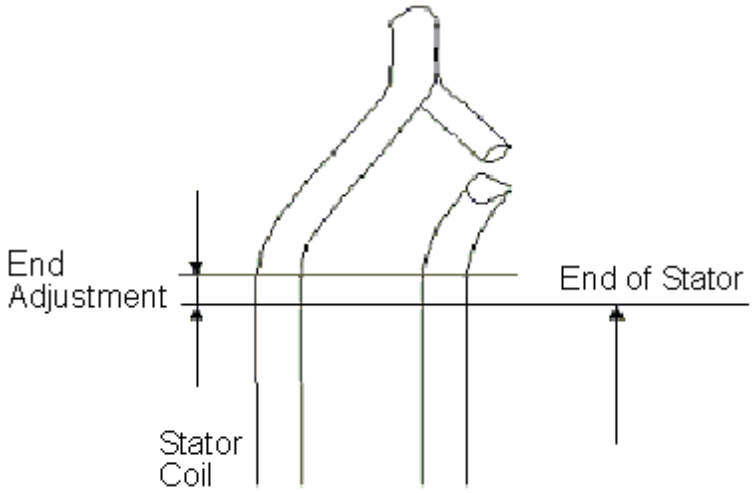
The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>General Options>Machines**command. When you are done setting the wire size, click **OK** to close the **Wire Size** window and return to the **Properties** window.

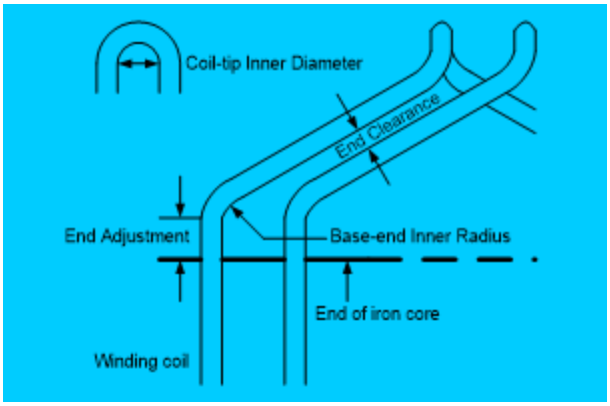
a. Click the **End/Insulation** tab.

b. Select or clear the **Input Half-turn Length** check box.

c. Do one of the following:

- If you selected **Input Half-turn Length**, then enter the half-turn length of the armature winding in the Half Turn Length field.
- If you cleared **Input Half-turn Length**, then enter the end length adjustment of the stator coils in the **End Adjustment** field. The end adjustment is the distance one end of the conductor extends vertically beyond the end of the stator.





5. Enter the inner radius of the base corner in the **Base Inner Radius** field.
6. Enter the inner diameter of the coil tip in the **Tip Inner Diameter** field.
7. Enter the distance between two stator coils in the **End Clearance** field.
8. Enter the thickness of the slot liner insulation in the **Slot Liner** field.
9. Enter the thickness of the wedge insulation in the **Wedge Thickness** field.
10. Enter the thickness of the insulation layer in the **Layer Insulation** field.
11. Enter the limited slot fill factor for the wire design in the **Limited Fill Factor** field. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.
12. Click **OK** to close the **Properties** window.

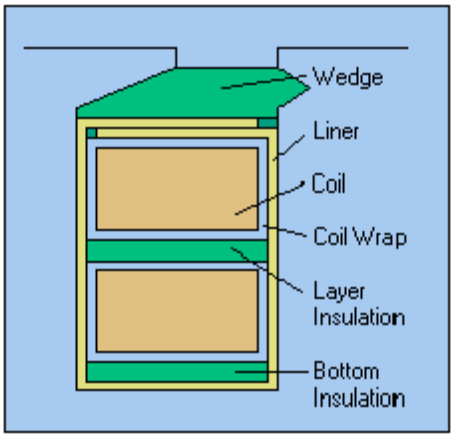
Stator Winding and Insulation for Three-Phase Synchronous Machines

To access the stator winding and insulation data, double-click the **Machine-Stator-Winding** entry in the project tree.

The **Stator Winding Data Properties** window contains the following fields:

Winding tab	**Winding Layers**	The number of winding layers.
	Winding Type	The type of stator winding. Click the button to open the **Winding Type** window and choose from **Whole Coiled**, **Half Coiled**, and **Editor**.
	Parallel Branches	The number of parallel branches in the stator winding.
	Conductors per Slot	The number of conductors per stator slot (0 for auto-design).
	Coil Pitch	The coil pitch measured in number of slots.
	Number of Strands	The number of wires per conductor (0 for auto-design).
	Wire Wrap	The thickness of the double-sided wire wrap (0 to automatically obtain this value from the wire library).
	Wire Size	The diameter of the wire (0 for auto-design). Click the button to open the **Wire Size** window where you can specify units, wire type, diameter, and gauge.
End/Insulation tab	**Input Half-turn Length7**	Select or clear this check box to specify whether or not you want to enter the half-turn length. When this check box is selected, the **Half Turn Length** field appears the next time you open the **Properties** window. When this check box is selected, the **End Adjustment** field appears instead.
	Half Turn Length	The half-turn length of the armature winding.
	End Adjustment	The end length adjustment of the stator coils, which is the distance one end of the conductor extends vertically beyond the end of the stator.
	Base Inner Radius	The inner radius of the base corner.

	Tip Inner Diameter	The inner diameter of the coil tip.
	End Clearance	The end clearance between two adjacent coils.
	Slot Liner	The thickness of the slot liner insulation.
	Wedge Thickness	The thickness of the wedge insulation.
	Layer Insulation	The thickness of the insulation layer.
	Limited Fill Factor	The limited slot fill factor for the wire design. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.



Winding Editor for a Three-Phase Synchronous Machine

For a three-phase synchronous machine, you may want to specify a different number of conductors for each stator slot. The Winding Editor makes this possible by enabling you to specify the number of turns for each coil.

To specify the number of turns for each coil:

1. Click **Machine>Winding>Edit Layout**.
   The **Winding Editor** window appears.
2. In the table in the upper left, set which phase you want for each coil and which slot is the “in” and “out” slot for the current in each coil.
3. If you are working on a quarter or half model, you may want to specify a multiplier by clicking the **Periodic Multiplier** check box and specifying a value.

4. Select or deselect the **Constant Turns** or **Constant Pitch** check boxes, depending on whether you want to be able to change these setting in the table above. When these options are selected, you cannot change the turns or pitch.
5. When you are satisfied with the coil settings, click **OK** to close the **Winding Editor** window.

Defining Different Size Wires for a Three-Phase Synchronous Machine

Use the **Gauge** option if you have a conductor that is made up different size wires.

To define different size wires:

1. In the **Wire Size** window, select **MIXED** from the **Gauge** pull-down menu.
2. Select either **Round** or **Rectangular** as the **Wire Type**.
3. Enter the appropriate wire data in the table:
   - For a round wire:
     - Enter the **Diameter** in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this diameter.
   - For a rectangular wire:
     - Enter the **Width** of the wire in the table.
     - Enter the **Thickness** of the wire in the table.
     - Enter the **Fillet** value in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this data.
4. Choose **Add** to add the new wire data.
5. Repeat steps 3 and 4 for each size wire you want to add.
6. When you are finished defining the wires, click **OK** to close the **Wire Size** window.

Note	For example, if one conductor is made up of 5 wires, and 3 of those wires have a diameter of 0.21mm, and the other 2 have a diameter of 0.13mm, then the mixed wire size table will have two lines. The first line will list Diameter = 0.21 and Number = 3. The second line will list Diameter = 0.13 and Number = 2. An equivalent wire diameter is displayed as Wire Size value in the Winding tab in the Properties window.

Stator Vent Data for Three-Phase Synchronous Machines

To insert a vent on a stator for a three phase synchronous machine:

1. Right click on the stator icon in the project tree to display the shortcut menu.
2. Click **Insert Vent**.

   The vent icon appears in the project tree under the stator.

To remove an existing vent item,

1. Right-click on the stator icon in the project tree to display the shortcut menu.
2. Click **Remove Vent**.

   This removes the vent item from the project tree.

To access the Vent properties for a vent, double click on a vent item. The **Vent Properties** window contains the following fields.

Vent Ducts	The number of radial vent ducts.
Duct Width	The width of the radial vent ducts.
Magnetic spacer width	Width of magnetic spacer which holds vent ducts. O for non-magnetic spacer.
Duct pitch.	Center-to-Center distance between two adjacent Vent ducts

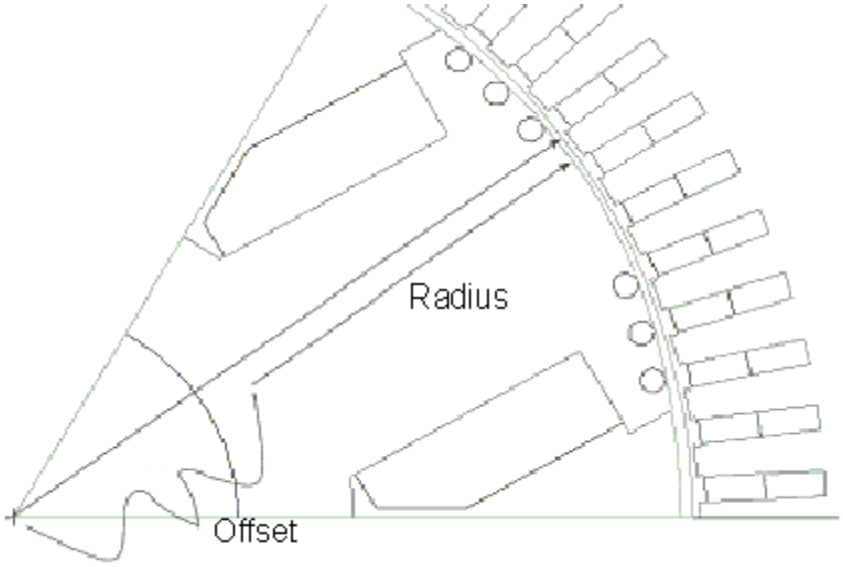
Defining the Rotor for a Three-Phase Synchronous Machine

The rotor consists of copper bars in which current is induced by the magnetic fields produced by the stator windings. In the project tree, double-click **Machine-Rotor** and **Machine-Rotor-Winding** to define the rotor.

To define the general rotor data:

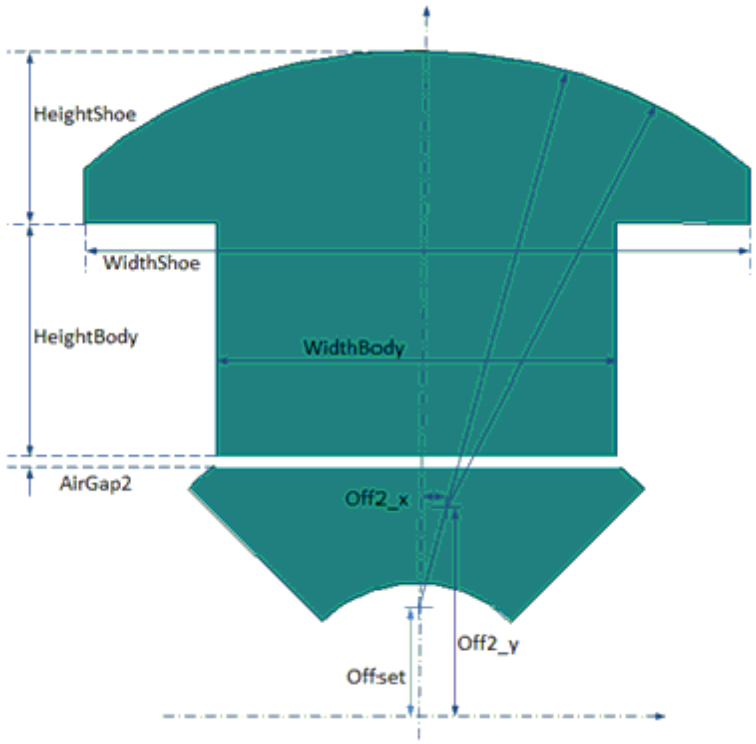
1. To open the **Rotor Data Properties** window, double-click the **Machine-Rotor** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Click the **Rotor** tab.
3. Enter the outer diameter of the rotor in the **Outer Diameter** field.
4. Enter the inner diameter of the rotor in the **Inner Diameter** field.
5. Enter the length of the rotor core in the **Length** field.
6. Select a **Steel Type** for the rotor core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
7. Enter the stacking factor for the rotor core in the **Stacking Factor** field.
8. Click the **Pole** tab.

9. Enter the pole-arc center offset from the rotor center in the **Pole Arc Offset** field.



10. Enter the width of the pole shoe in the **Pole Shoe Width** field.
11. Enter the height of the pole shoe in the **Pole Shoe Height** field.
12. Enter the width of the pole body in the **Pole Body Width** field.
13. Enter the height of the pole body in the **Pole Body Height** field.
14. Enter the width between the rotor pole and rotor yoke in the **Second Air Gap** field.
15. To include the two arcs in the half-pole range, do the following:
    a. Select the **Select Pole Arc** check box.
    b. Enter the offset of the second arc perpendicular to the pole-center line in the **Off2_x** field.
    c. Enter the offset of the second arc parallel with the pole-center line in the **Off2_y** field.

Note	If the Off2_x and Off2_y parameters for the second arc are invalid, the second arc is ignored and the pole includes only the first arc defined by Pole Arc Offset.



1. Select or clear the **Magnetic PressBoard** check box to specify whether or not the press board is made of magnetic material.
2. Enter the thickness of the press board in the **Press Board Thickness** field.
3. Click the **Insulation** tab.
4. Enter the thickness of the insulating material beneath the shoe pole in the **Shoe Insulation** field.
5. Enter the thickness of the insulating material on the side of the pole body in the **Pole Insulation** field.
6. Enter the clearance distance between the windings in the **Winding Clearance** field.
7. Click **OK** to close the **Properties** window.

Rotor, Rotor Pole, and Insulation for Three-Phase Synchronous Machines

To access the general rotor data, pole data, and insulation data double-click the **Machine>Rotor** entry in the project tree.

The **Rotor Data Properties** window contains the following fields:

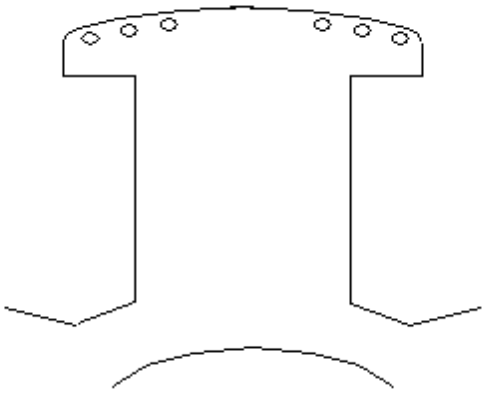
Rotor tab	**Outer Diameter**	The outer diameter of the rotor core.
	Inner	The inner diameter of the rotor core.

	Diameter	
	Length	The length of the rotor core.
	Steel Type	The steel type of the rotor core. Click the button to open the **Select Definition** window.
	Stacking Factor	The stacking factor of the rotor core.
Pole tab	**Pole Arc Offset**	The pole-arc center offset from the rotor center.
	Pole Shoe Width	The width of the pole shoe.
	Pole Shoe Height	The height of the pole shoe.
	Pole Body Width	The width of the pole body.
	Pole Body Height	The height of the pole body.
	Second Air Gap	The width of the second air gap, between the rotor pole and rotor yoke.
	Second Pole Arc	Select or clear this option to specify whether or not the pole surface includes the two arcs in the half-pole range. When you select this check box, two additional fields appear: **Off2_x** and **Off2_y**.
	Off2_x	The offset of the second arc perpendicular to the pole-center line. This field is only available when **Second Pole Arc** is selected. **Note:** If **Off2_x** for the second arc is invalid, the second arc is ignored and the pole includes only the first arc defined by **Pole Arc Offset**.
	Off2_y	The offset of the second arc parallel with the pole-center line. This field is only available when **Second Pole Arc** is selected. **Note:** If **Off2_y** for the second arc is invalid, the second arc is ignored and the pole includes only the first arc defined by **Pole Arc Offset**.
	Magnetic PressBoard	Select or clear this option to specify whether or not the press board is made of magnetic material.
	Press Board Thickness	The thickness of the press board.
	Steel Type	The steel type of the rotor pole. Click the button to open the **Select Definition** window. Default value is the same as the rotor core.
	Stacking Factor	The stacking factor of the rotor pole. Default value is the same as the rotor core.

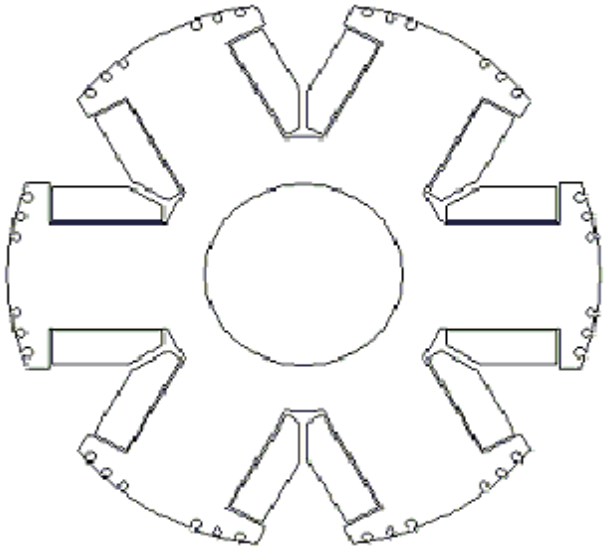
Insulation tab	**Shoe Insulation**	The thickness of the insulating material beneath the pole shoe.
	Pole Insulation	The thickness of the insulating material on the side of the pole body.
	Winding Clearance	The clearance distance between the windings.

Defining the Rotor Pole for a Three-Phase Synchronous Machine

The rotor pole drives the electromagnetic field that is coupled with the stator windings.
The following figure shows a partial diagram of a rotor pole:

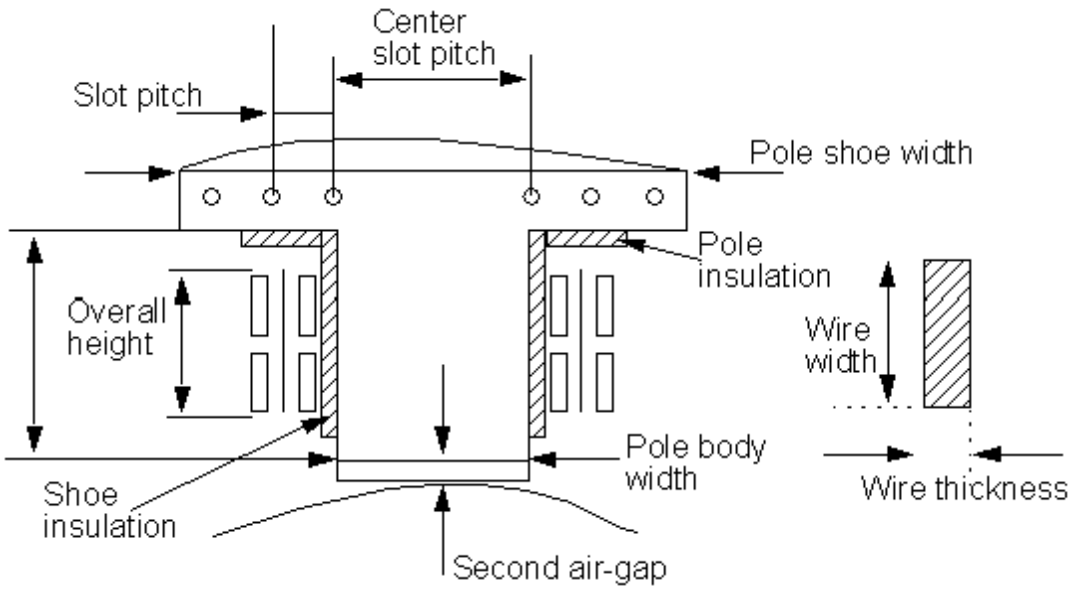


The following figure shows a diagram of an entire rotor:



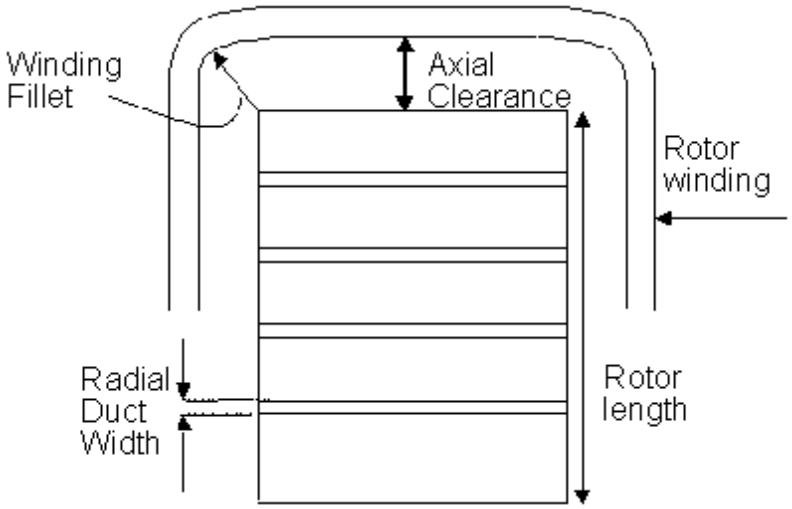
Defining the Rotor Winding Data for a Three-Phase Synchronous Machine

Use the **Rotor Winding** window to define the wires and physical dimensions of the rotor winding. The rotor winding provides the excitation for the electromagnetic field that produces the rotor pole.



To define the rotor windings:

1. To open the **Rotor Winding Properties** window, double-click the **Machine-Rotor-Winding** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select the **Winding Type** for the rotor:
   a. Click the button.
      The **Winding Type** window appears.
   b. Click to select the type of winding, from **Round**, **Cylinder**, or **EdgeWise**.
      When you place the mouse cursor over the winding type, a schematic of the selected winding appears
   c. Click **OK** to return to the **Properties** window.
3. Enter the number of parallel branches for the winding in the **Parallel Branches** field.
4. Conductors per Pole
5. Enter the number of wires in each conductor in the **Number of Strands** field.
6. Enter the width of the insulating wire wrap in the **Wire Wrap** field.
7. Interturn Insulation
8. Enter the gauge of the wire in the **Wire Size** field.
9. Enter the **Axial Clearance** to specify the axial distance between the core and the coil at the end of the lamination stack.



10. Limited Cross Width
11. Limited Cross Height
12. Winding Fillet
13. Click **OK** to close the **Properties** window.

Rotor Winding Data for Three-Phase Synchronous Machines

To access the rotor winding data, double-click the **Machine-Rotor-Winding** entry in the project tree.

The **Rotor Winding Data Properties** window contains the following fields:

Winding Type	The type of rotor winding. Click the button to open the **Winding Type** window and choose from **Whole Coiled**, **Half Coiled**, and **Editor**.
Parallel Branches	The number of parallel branches in the rotor winding.
Conductors per Pole	The number of conductors per rotor pole (0 for auto-design).
Number of Strands	The number of wires per conductor (0 for auto-design).
Wire Wrap	The thickness of the double-sided wire wrap (0 to automatically obtain this value from the wire library).
Interturn Insulation	The thickness of the inter-turn insulation of an edgewise winding. This field only appears when **EdgewiseCoil** is selected as the **Winding Type**.
Wire Size	The diameter of the wire (0 for auto-design). Click the button to open the **Wire Size** window where you can specify units, wire type, diameter, and gauge.
Axial Clearance	The axial gap between the field winding and the pole body or inner coil.

Limited Cross Width	The limited cross-section width for the winding design or arrangement (0 for available maximum area).
Limited Cross Height	The limited cross-section height for the winding design or arrangement (0 for available maximum area).
Winding Fillet	The size of the winding fillet.

Defining the Rotor Damper Data

To define a rotor damper for a machine that permits one:

1. Click **Machine>Insert Damper.**

   The Damper icon appears in the project tree under the rotor icon. A slot icon appears in the hierarchy under the damper.
2. Double click on the Damper icon to display the properties window for the damper.
3. Enter the appropriate values for the damper. The slot type, the bar conductor type, and end conductor type are entered by clicking on buttons that open other windows.
4. Click OK to close the properties window.

Damper Data for Three-Phase Synchronous Machines

By option, you can add a damper to or remove damper from the rotor of a three phase machine.

To add a damper:

1. Right-click on the rotor icon in the project tree to display the short cut menu.
2. Click **Insert Damper** on the menu.

   The damper appears in the project tree under the rotor. The damper also includes an associated slot.

1. To remove a damper, right-click on the rotor icon in the project tree to display the short cut menu.
2. Click **Remove Damper** on the menu.

   The damper and associated slot are removed from the project tree.

The damper data contains the following fields.

Damper slots per pole	Number of damper slots per pole.
Slot type	Damper slot type. Specify this by clicking the button in the properties field and selecting from the Select Slot Type window.
Cast Rotor	Whether the rotor squirrel cage winding is cast.

Bar conductor type	Specify this by clicking the button in the properties field, and using the Select Definition window to find and assign materials.
End length	Single side end extended bar length/
End ring width	Axial width of end ring.
End ring height	Radial height of end ring.
End ring conductor type	Specify this by clicking the button in the properties field and using the Select Definition window to find and assign the material.
End Ring type	Type of end ring for the damper. Specify this by clicking the button in the properties field and use the Select Pole type window to select from the available types.
Slot pitch	Slot pitch in mechanical degrees in reference to the offset center.

Defining the Shaft Data for a Three-Phase Synchronous Machine

To define the shaft:

1. To open the **Shaft Data Properties** window, double-click the **Machine>Shaft** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select or clear the **Magnetic Shaft** check box to specify whether or not the shaft is to be made of magnetic material.
3. Click **OK** to close the **Properties** window.

Shaft Data for Three-Phase Synchronous Machines

To access the shaft data, double-click the **Machine>Shaft** entry in the project tree.

The **Shaft Data Properties** window contains the following fields:

Magnetic Shaft	Select or clear this check box to indicate whether or not the shaft is made of magnetic material.

Setting Up Analysis Parameters for a Three-Phase Synchronous Machine

To define the solution data:

1. To open the **Solution Setup** window, right-click **Analysis** in the project tree, and click **Add Solution Setup**.
2. Click the **General** tab.

3. Select **Motor** or **Generator** from the **Operation Type** pull-down list.
4. Select the **Load Type** used in the motor from the following options:

Infinite Bus	When the generator is connected to an infinite bus of the power system, the terminal voltages remain constant and the output power is determined by the output currents.
Independent Generator	The generator supplies electrical power to a load inductor. For an AC generator, the resistance and inductance of the load inductor are determined by the rated output power, the rated voltage, and the rated power factor. For a DC generator, the resistance of the load resistor is determined by the rated output power and the rated voltage.
Const Speed	*For Motors.* The speed remains constant in the motor.
Const Power	*For Motors.* The output power remains constant in the motor.
Const Torque	*For Motors.* The torque remains constant regardless of the speed. In this case, $T_{load} = T_{rated}$, given by the output power divided by the given rated speed.
Linear Torque	*For Motors.* The torque increases linearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})$ where T_{rated} is given by the output power divided by the given rated speed.
Fan Load	*For Motors.* The load varies nonlinearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})^2$ where T_{rated} is given by the output power divided by the given rated speed.

5. Enter the output power developed at the shaft of the machine in the **Rated Output Power** field.
6. Enter the RMS line-to-line voltage in the **Rated Voltage** field.
7. Enter the desired output speed of the motor at the load point in the **Rated Speed** field.
8. Enter the temperature at which the system functions in the **Operating Temperature** field.
9. Click the **Three-Phase Synchronous Machine** tab.
10. Enter a value in the **Rated Power Factor** field.
11. Select **Wye** or **Delta** from the **Winding Connection** pull-down list.
12. In the **Exciter Efficiency** field, enter the efficiency of the exciter used to supply the rotor winding with DC current if it is mechanically connected to the shaft of the generator. The efficiency value ranges between 0 and 1 and will only affect the total efficiency result.
13. To enter an **Input Exciting Current**, select the check box, enter a value, and select the units.
14. Click **OK** to close the **Solution Setup** window.

Related Topics

Solution Data for Three-Phase Synchronous Machines

Solution Data for Three-Phase Synchronous Machines

To access the solution data, right-click **Analysis** in the project tree, and click **Add Solution Setup**.

The **Solution Setup** window contains the following fields:

Operation Type	*On the* **General** *tab.* Select from **Motor** or **Generator**.
Load Type	*On the* **General** *tab.* For a motor, select from **Const Speed**, **Const Power**, **Const Torque**, **Linear Torque**, and **Fan Load**. The default is **Const Power**. For a generator, select from **Infinite Bus** and **Independent Generato**r.
Rated Output Power	*On the* **General** *tab.* Type a value for the rated output voltage, and select the units.
Rated Voltage	*On the* **General** *tab.* Type a value for the rated voltage, and select the units.
Rated Speed	*On the* **General** *tab.* Type a value for the rated speed, and select the units.
Operating Temperature	*On the* **General** *tab.* Type a value for the operating temperature, and select the units.
Rated Power Factor	*On the* **Three-Phase Synchronous Machine** *tab.* Type a value for the rated power factor.
Winding Connection	*On the* **Three-Phase Synchronous Machine** *tab.* Select from **Wye** or **Delta**.
Exciter Efficiency	*On the* **Three-Phase Synchronous Machine** *tab.* Type a percent for the exciter efficiency.
Input Exciting Current	*On the* **Three-Phase Synchronous Machine** *tab.* If you select this check box, then enter the exciting current, and select the units.

Related Topics

Setting Up Analysis Parameters for a Three-Phase Synchronous Machine

Brushless Permanent-Magnet DC Motors

After you have selected **Brushless Permanent-Magnet DC Motors** as your model type, you need to define the following:

- General data, such as the voltage, speed, and circuit type of the model.
- Circuit data, such as lead trigger angle, transistor drop, and control circuit information.
- Stator data, such as the diameter, slot dimensions, winding data, and skew width of the stator.
- Rotor data
- Rotor pole data, such as the magnet dimensions and stacking factor.

- Shaft data
- Optional machine housing.
- Solution data, such as rated output voltage and frequency.

Analysis Approach for Brushless PMDC Motors

The stator of a brushless DC motor is equipped with a polyphase winding. The phases are connected to the DC bus through a switching circuit. The switching sequence is controlled so that it is synchronized with the position of the rotor. As a result, the stator produces a rotating magnetic field.

The rotor is equipped with permanent magnets, creating a structure with the same number of poles at the stator. The stator switches act like a commutator in a classic DC motor.

In brushless permanent-magnet DC (BLDC) motors, the armature currents are commutated exactly according to rotor position. The signal of rotor position may be obtained from a position sensor, or from induced voltages for sensor-less control system.

The performance of BLDC motors is analyzed via a time-domain simulation. The voltage equation in the time domain is:

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} - \begin{bmatrix} e_d \\ e_q \\ e_0 \end{bmatrix} = \begin{bmatrix} R_1 + L_d p & -L_q \omega_e & 0 \\ -L_d \omega_e & R_1 + L_q p & 0 \\ 0 & 0 & R_1 + L_0 p \end{bmatrix} \cdot \begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix}$$

where R_1, L_d, L_q, and L_0 are armature resistance, d-axis synchronous inductance, q-axis synchronous inductance, and 0-axis inductance, respectively. ω_e is rotor speed in electrical rad/s, and p represents for *d/dt*.

The transformations for terminal voltages, induced voltages, and winding currents are given by the following three equations:

$$\begin{bmatrix} v_d \\ v_q \\ v_0 \end{bmatrix} = C^T \cdot \begin{bmatrix} v_a \\ v_b \\ ? \end{bmatrix} : \quad \begin{bmatrix} e_d \\ e_q \\ e_0 \end{bmatrix} = C^T \cdot \begin{bmatrix} e_a \\ e_b \\ ? \end{bmatrix} \quad : \begin{bmatrix} i_a \\ i_b \\ ? \end{bmatrix} = C \cdot \begin{bmatrix} i_d \\ i_q \\ i_0 \end{bmatrix}$$

The transformation matrices for 2-phase, 3-phase, and 4-phases systems, noted as C_2, C_3, and C_4, are as follows:

$$C_2 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \end{bmatrix}$$

$$C_3 = \sqrt{\frac{2}{3}} \begin{bmatrix} \cos\theta & \sin\theta & 1/(\sqrt{2}) \\ \cos(\theta-\alpha) & \sin(\theta-\alpha) & 1/(\sqrt{2}) \\ \cos(\theta-2\alpha) & \sin(\theta-2\alpha) & 1/(\sqrt{2}) \end{bmatrix}$$

$$C_4 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ \sin\theta & -\cos\theta & 0 \\ -\cos\theta & -\sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \end{bmatrix}$$

where $\alpha = 2\pi/3$.

The input power (electric power) can now be computed from the voltage and current as:

$$p_1 = \frac{1}{t}\int_0^T (v_d i_d + v_q i_q + v_0 i_0)dt$$

The output power (mechanical power) is:

$$P_2 = P_1 - (P_{fw} + P_{Cua} + P_t + P_{Fe})$$

where P_{fw}, P_{Cua}, P_t, and P_{Fe} are frictional and wind loss, armature copper loss, transistor/diode loss, and iron-core loss, respectively.

The output mechanical shaft torque T_2 is:

$$T_2 = P_2 / \omega$$

where ω is the rotor speed in mechanical rad/s.

The efficiency is computed by:

$$eff = P_2/P_1 * 100\%$$

Defining a Brushless Permanent-Magnet DC Motor

The general procedure for defining a brushless permanent-magnet DC motor is as follows:

1. Insert the permanent magnet brushless DC motor into a new or existing project.
2. Double-click the **Machine** entry in the project tree to define the general data.
3. Double-click the **Machine-Circuit** entry in the project tree to define the control circuit.
4. Double-click the **Machine-Stator** entry in the project tree to define the stator geometry.
5. Double-click the **Machine-Stator-Slot** entry in the project tree to define the stator slot dimensions.
6. Double-click the **Machine-Stator-Winding** entry in the project tree to define the stator windings and conductors.
7. Double-click the **Machine-Rotor** entry in the project tree to define the rotor geometry.
8. Double-click the **Machine-Rotor-Pole** entry in the project tree to define the pole, embrace, offset, and air gap data for the rotor pole.
9. Double-click the **Machine-Shaft** entry in the project tree to define the magnetism of the shaft.
10. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
11. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.

12. Choose **File>Save** to save the project.
13. Choose **RMxprt>Analyze** to analyze the design.

Note	When you place the cursor over an entry field in the data windows, a brief description of that field appears in the status bar at the bottom of the **RMxprt** window.

Once analyzed, the model can be viewed in the Maxwell 2D Modeler, or it can be used to create a new Maxwell 2D project, and a new Maxwell 3D design

Please refer to the *Brushless Permanent-Magnet DC Motor Problem* application note, on the technical support page of the Ansys web site, for a specific example of a brushless permanent-magnet DC motor problem.

Defining the General Data for a Brushless PMDC Motor

Use the **General** window to specify the rated output power, voltage values, circuit type, and speed of the brushless DC motor.

To define the general data:

1. To open the **General Data Properties** window, double-click the **Machine** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the number of poles for the machine in the **Number of Poles** field. This value is the total number of poles in the stator (or the number of pole pairs multiplied by two).
3. Enter the energy loss due to friction at the given speed in the **Frictional Loss** field.
4. Enter the wind loss due to air resistance measured at the reference speed in the **Wind Loss** field.
5. Enter the given speed in the **Reference Speed** field.
6. Select **DC** or **CCC** from the **Control Type** pull-down list.
7. Select a **Circuit Type** from the following types:

Y3	Y-connected, three-phase.
L3	Loop-type, three-phase.
S3	Star-type, three-phase.
C2	Cross-type, two-phase.
L4	Loop-type, four-phase.
S4	Star-type, four-phase.

   The circuit types are based on industry standards. By default, type **Y3**, a three-phase, six-status circuit, is selected as the circuit type.

Note	When you place the mouse cursor over a circuit type, an outline schematic of the circuit appears.

8. Click **OK** to close the **Properties** window.

General Data for Brushless PMDC Motors

To access the general data, double-click the **Machine** entry in the project tree.

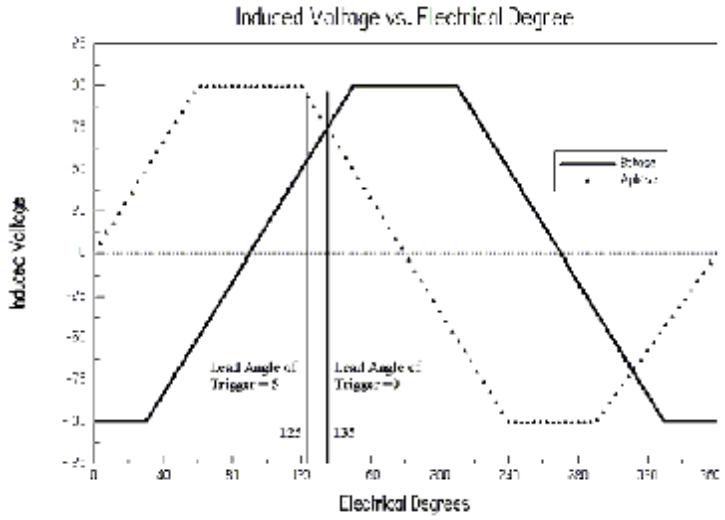
The **General Data Properties** window for a three-phase induction motor contains the following fields:

Machine Type	The machine type you selected when inserting a new RMxprt design (Brushless Permanent-Magnet DC Motor).
Number of Poles	The number of poles the machine contains.
Rotor Position	Select whether the rotor is an **Inner Rotor** or **Outer Rotor**.
Frictional Loss	The frictional energy loss (due to friction) measured at the reference speed.
Wind Loss	The wind loss (due to air resistance) measured at the reference speed.
Reference Speed	The given speed of reference.
Control Type	The way the circuit is controlled. Select from **DC** or **CCC** (chopped current control).
Circuit Type	The drive circuit type. Click the button to open the **Circuit Type** window and select from the following six types: • **Y3**: Y-Type, 3-Phase • **L3**: Loop-Type, 3-Phase • **S3**: Star-Type, 3-Phase • **C2**: Cross-Type, 2-Phase • **L4**: Loop-Type, 4-Phase • **S4**: Star-Type, 4-Phase

Defining the Circuit Data for a Brushless PMDC Motor

Use the **Circuit Data Properties** window to define the circuit data for a brushless PMDC Motor.

1. To open the **Circuit Data Properties** window, double-click the **Machine>Circuit** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the trigger's lead angle in electrical degrees in the **Lead Angle of Trigger** field. The trigger's lead angle is shown in the following plot of the open circuit induced voltage versus position. An angle of 0 means that the induced voltage in the triggered phase is at a maximum:



Note	A positive value represents a lead angle, and a negative value represents a lag angle.

3. Enter the period from on-status to off-status of a transistor, in electrical degrees, in the **Trigger Pulse Width** field.
4. Enter the voltage drop across one transistor when the transistor is turned on in the **Transistor Drop** field. Refer to the figures of the different circuit types in step 2.
5. Enter the voltage drop of one diode in the discharge loop in the **Diode Drop** field. If you selected a star-type circuit (**S3** or **S4**) as the **Circuit Type**, enter the total discharge voltage in this field.
6. If you selected CCC (chopped current control) as the **Control Type**, then enter the maximum and minimum current values in the **Maximum Current** and **Minimum Current** fields.
7. Click **OK** to close the **Properties** window.

Circuit Data for Brushless PMDC Motors

To access the **Circuit Data Properties** window, double-click the **Machine>Circuit** entry in the project tree.

Lead Angle of Trigger	The trigger's lead angle, in electrical degrees.
Trigger Pulse Width	The period from on-status to off-status for a transistor, in electrical degrees.
Transistor Drop	The voltage drop across one transistor when the transistor is turned on.
Diode Drop	The voltage drop across one diode in the discharge loop.
Maximum	The maximum current for the chopped current control. This field is not

Current	available for a DC circuit.
Minimum Current	The minimum current for the chopped current control. This field is not available for a DC circuit.

Defining the Stator Data for a Brushless PMDC Motor

The stator is the outer lamination stack where the polyphase voltage windings reside.

To define the general stator data:

1. To open the **Stator Data Properties** window, double-click the **Machine>Stator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the **Outer Diameter** of the stator.
3. Enter the **Inner Diameter** of the stator.
4. Enter the length of the stator core in the **Length** field.
5. Enter the stacking factor for the stator core in the **Stacking Factor** field.
6. Select a **Steel Type** for the stator core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
7. Enter the **Number of Slots** in the stator.
8. Select the **Slot Type**:
   a. Click the button for the **Slot Type**.
      The **Select Slot Type** window appears.
   b. Select a slot type (available types include 1 through 4).

Note	When you place the mouse cursor over the slot type, a schematic of the selected type appears, displaying the slot dimension variables.

   c. Click **OK** to close the **Select Slot Type** window and return to the **Properties** window.
9. Enter the skew width, measured in slot number, in the **Skew Width** field.
10. Click **OK** to close the **Properties** window.

Stator Data for Brushless PMDC Motors

To access the general stator data, double-click the **Machine>Stator** entry in the project tree.

The **Stator Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the stator core.
Inner Diameter	The inner diameter of the stator core.

Length	The length of the stator core.
Stacking Factor	The stacking factor of the stator core.
Steel Type	The steel type of the stator core. Click the button to open the **Select Definition** window.
Number of Slots	The number of slots the stator core contains.
Slot Type	The type of slots in the stator core. Click the button to open the **Select Slot Type** window.
Skew Width	The skew width measured in slot number.

Defining the Stator Slots for a Brushless PMDC Motor

To define the physical dimensions of the stator slots:

1. To open the **Stator Slot Data Properties** window, double-click the **Machine-Stator-Slot** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Optionally, to automatically design the dimensions of slots **Hs2**, **Bs1**, and **Bs2**, select the **Auto Design** check box.
3. Optionally, to design dimensions of slots **Bs1** and **Bs2** based on the stator tooth width, select the **Parallel Tooth** check box, and enter a value in the **Tooth Width** field.
4. Enter the available slot dimensions.

Hs0	Always available.
Hs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.
Bs0	Always available.
Bs1	Available only when **Auto Design** and **Parallel Tooth** are both cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Bs2	Available only when **Auto Design** and **Parallel Tooth** are both cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.

5. Click **OK** to close the **Properties** window.

Stator Slot Data for Brushless PMDC Motors

To access the stator slot data, double-click the **Machine-Stator-Slot** entry in the project tree.

The **Stator Slot Data Properties** window contains the following fields:

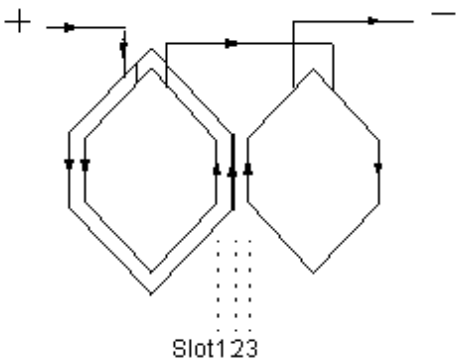
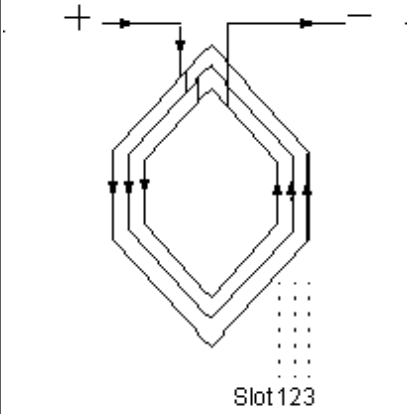
Auto Design	Select or clear this to enable or disable auto-design of slots **Hs2**, **Bs1**, and **Bs2**. When this check box is selected, only two other fields appear in the window: **Hs0** and **Bs0**.
Parallel Tooth	Select this to design **Bs1** and **Bs2** based on the tooth width. When this check box is selected, the **Bs1** and **Bs2** fields are removed, and the **Tooth Width** field is added.
Tooth Width	The tooth width for the parallel tooth, on which **Bs1** and **Bs2** are designed.
Hs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Hs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Hs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Rs	A slot dimension. (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected). Rs is added when the slot type is 3 or 4.

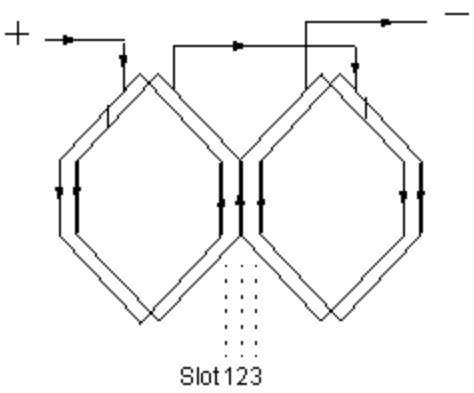
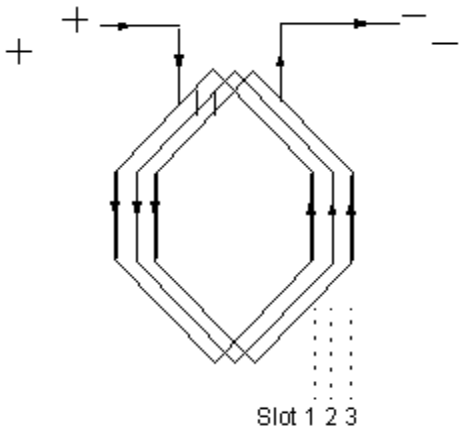
Defining the Stator Windings and Conductors for a Brushless PMDC Motor

To define the stator windings, wires, and conductors:

1. To open the **Stator Slot Winding Properties** window, double-click the **Machine-Stator-Winding** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Click the **Winding** tab.
3. Enter the number of layers in the stator winding in the **Winding Layers** field.
4. Select a **Winding Type**:
   a. Click the button for **Winding Type**.
   The **Winding Type** window appears.
   b. Select from one of the following three types of winding:
      - **Whole Coiled**
      - **Half Coiled**
      - **Editor**

When you place the mouse cursor over a winding button, an outline of the selected winding appears. The following table describes the six types of windings that are possible (three for one-layer and three for two-layer):

Type	Description
Editor	A user-defined one-layer winding arrangement. You need to set up the winding arrangement for each slot. For this winding type, the following letters are used for the phase windings: • Phase A/A return uses A/X. • Phase B/B return uses B/Y. • Phase C/C return uses C/Z.
Whole Coiled	A one-layer whole-coiled winding: 
Half Coiled	A one-layer concentric half-coiled winding: 
Editor	A user-defined two-layer winding arrangement. When you select **20**, the Winding Editor opens, where you can specify a different winding arrangement for each slot.
Whole Coiled	A two-layer wave winding:

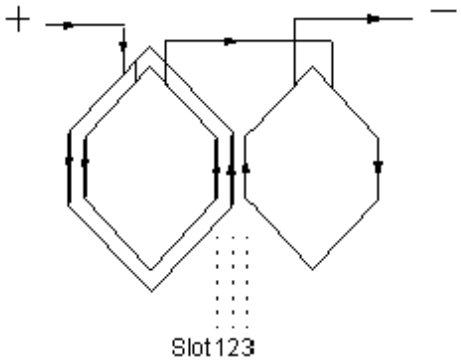
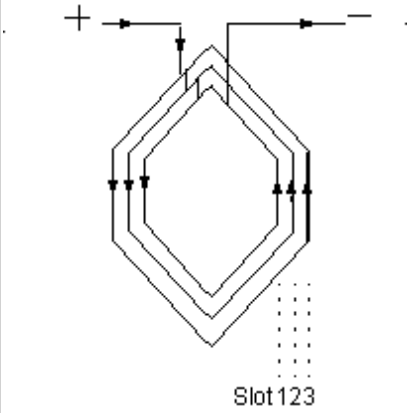
	 The phase belt for this winding configuration is equal to 360/2*m*, where *m* is the phase number.
Half Coiled	A two-layer half-coiled winding:  There is only one coil per phase per pair of poles.

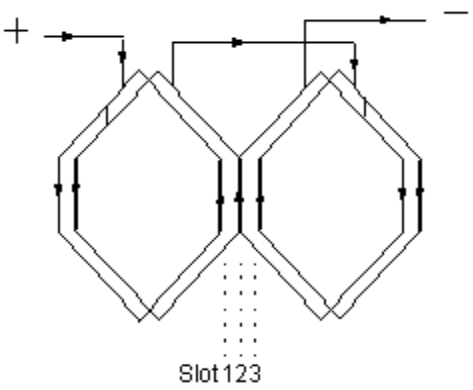
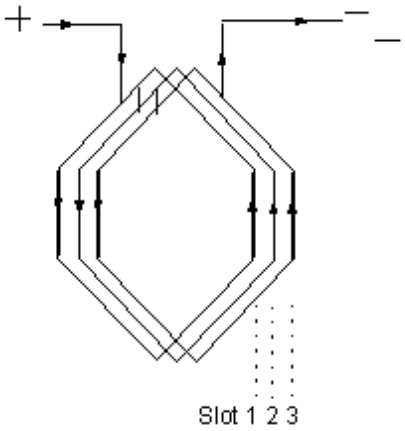
Note	**For a two layer winding,** if you check Constant Pitch in the Winding Editor, only the top layer needs to be defined; the bottom layer will be determined according to the coil pitch.

Once you have clicked a button to select a winding, click **OK** to close the **Winding Type** window and return to the **Properties** window.

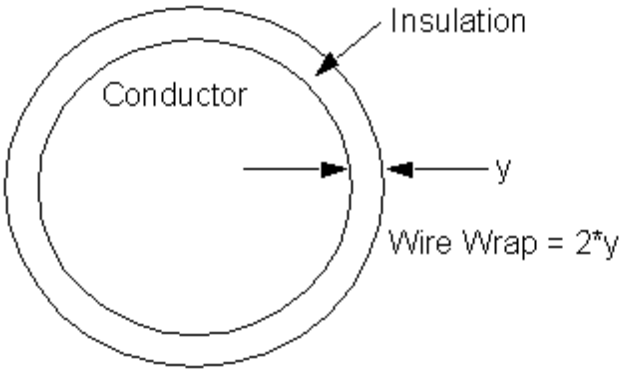
Select a **Winding Type**. The following winding types are available:

Note	When you place the mouse cursor over a winding, an outline of the selected winding appears.

10	A user-defined single-layer winding arrangement. When you select this type, enter the winding arrangement, and choose **OK**. For this winding type, the following letters are used for the phase windings: • Phase A/A return uses A/X. • Phase B/B return uses B/Y. • Phase C/C return uses C/Z.
11	A one-layer whole-coiled winding: 
12	A one-layer concentric half-coiled winding: 
20	A user-defined two-layer winding arrangement. When you select this type, enter the winding arrangement, and choose **OK**.
21	A two-layer wave winding:

	 The phase belt for this winding configuration is equal to 360/2*m*, where *m* is the phase number.
22	A two-layer winding: 

a. Select or enter the number of parallel branches in one phase of the winding in the **Parallel Branches** field.
b. Enter the total number of conductors in each stator slot in the **Conductors per Slot** field. This value is the number of turns per coil multiplied by the number of layers.
c. Enter the coil pitch, measured in number of slots, in the **Coil Pitch** field. The coil pitch is the number of slots separating one winding. For example, if a coil starts in slot 1 and ends in slot 6, it has a coil pitch of 5.
d. Enter the number of wires per conductor in the **Number of Strands** field. Enter 0 to have RMxprt auto-design this value.
e. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



f. Select the **Wire Size**:
   g. Click the button for **Wire Size**.
   The **Wire Size** window appears.
   h. Select a value from the **Wire Diameter** pull-down list.
   i. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

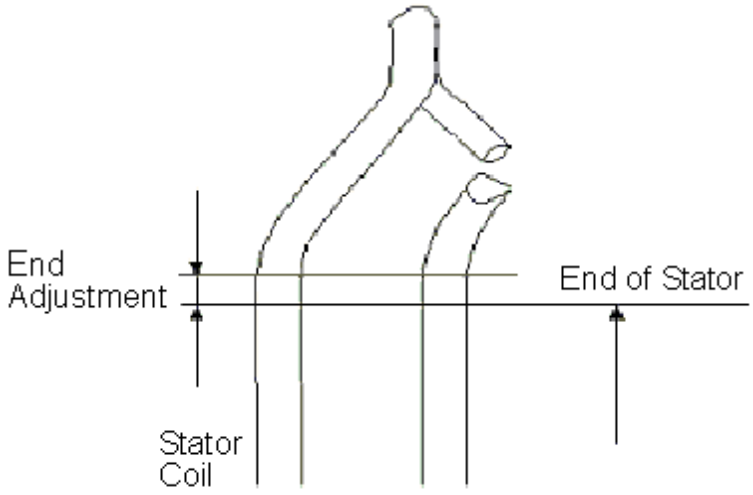
< ***number*** **>**	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.
USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.
AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.
MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21mm and 2 with a diameter of 0.13mm.

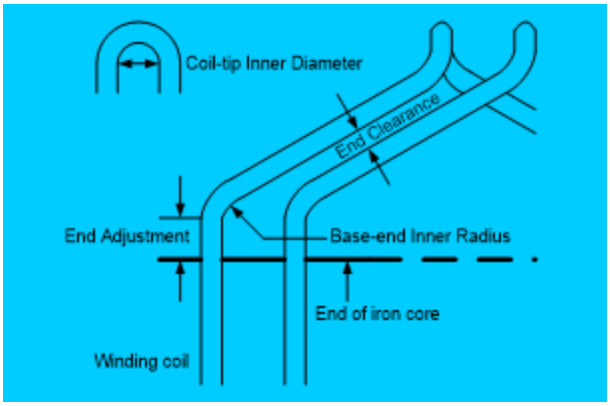
The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>General Options>Machines**command.

When you are done setting the wire size, click **OK** to close the **Wire Size** window and return to the **Properties** window.

a. Click the **End/Insulation** tab.
b. Select or clear the **Input Half-turn Length** check box.
c. Do one of the following:
   - If you selected **Input Half-turn Length**, then enter the half-turn length of the armature winding in the Half Turn Length field.

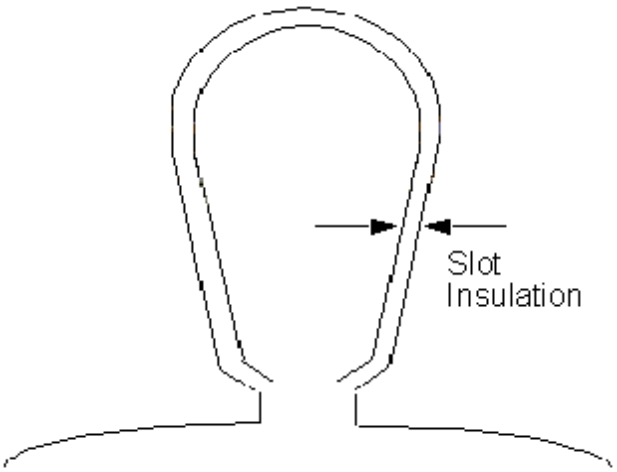
- If you cleared **Input Half-turn Length**, then enter the end length adjustment of the stator coils in the **End Adjustment** field. The end adjustment is the distance one end of the conductor extends vertically beyond the end of the stator.





5. Enter the inner radius of the base corner in the **Base Inner Radius** field.
6. Enter the inner diameter of the coil tip in the **Tip Inner Diameter** field.
7. Enter the distance between two stator coils in the **End Clearance** field.

8. Enter the thickness of the slot liner insulation in the **Slot Liner** field.



9. Enter the thickness of the wedge insulation in the **Wedge Thickness** field.
10. Enter the thickness of the insulation layer in the **Layer Insulation** field.
11. Enter the limited slot fill factor for the wire design in the **Limited Fill Factor** field. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.
12. Click **OK** to close the **Properties** window.

Winding Editor for a Brushless DC Motor

For a brushless DC motor, you may want to specify a different number of conductors for each stator slot. The Winding Editor makes this possible by enabling you to specify the number of turns for each coil.

To specify the number of turns for each coil:

1. Click **Machine>Winding>Edit Layout**.
   The **Winding Editor** window appears.
2. In the table in the upper left, set which phase you want for each coil and which slot is the “in” and “out” slot for the current in each coil.
3. If you are working on a quarter or half model, you may want to specify a multiplier by clicking the **Periodic Multiplier** check box and specifying a value.
4. Select or deselect the **Constant Turns** or **Constant Pitch** check boxes, depending on whether you want to be able to change these setting in the table above. When these options are selected, you cannot change the turns or pitch.
5. When you are satisfied with the conductor settings, click **OK** to close the **Winding Editor** window.

Defining Different Size Wires for a Brushless DC Motor

Use the **Gauge** option in the **Wire Size** dialog if you have a conductor that is made up different size wires.

To define different size wires:

1. In the **Wire Size** window, select **MIXED** from the **Gauge** pull-down menu.
2. Select either **Round** or **Rectangular** as the **Wire Type**.
3. Enter the appropriate wire data in the table:
   - For a round wire:
     - Enter the **Diameter** in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this diameter.
   - For a rectangular wire:
     - Enter the **Width** of the wire in the table.
     - Enter the **Thickness** of the wire in the table.
     - Enter the **Fillet** value in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this data.
4. Click **Add** to add the new wire data.
5. Repeat steps 3 and 4 for each size wire you want to add.
6. When you are finished defining the wires, click **OK** to close the **Wire Size** window and return to the RMxprt **Properties** window.

Note	For example, if one conductor is made up of 5 wires, and 3 of those wires have a diameter of 0.21mm, and the other 2 have a diameter of 0.13mm, then the mixed wire size table will have two lines. The first line will list Diameter = 0.21 and Number = 3. The second line will list Diameter = 0.13 and Number = 2. An equivalent wire diameter is displayed as Wire Size value in the Winding tab in the Properties window.

Stator Winding Data for Brushless PMDC Motors

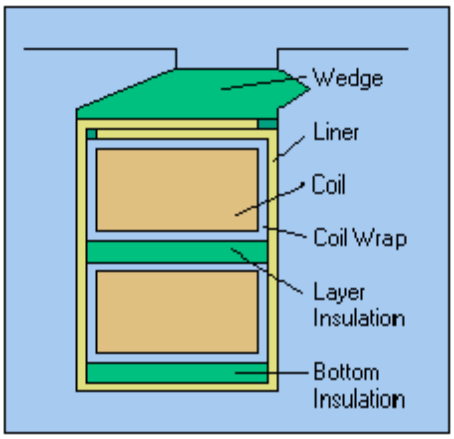
To access the stator winding data, double-click the **Machine-Stator-Winding** entry in the project tree.

The **Stator Winding Data Properties** window contains the following fields:

Winding tab	**Winding Layers**	The number of winding layers.
	Winding Type	The type of stator winding. Click the button to open the **Winding Type** window and choose from **Whole Coiled**, **Half Coiled**, and **Editor**.
	Parallel Branches	The number of parallel branches in the

		stator winding.
	Conductors per Slot	The number of conductors per stator slot (0 for auto-design).
	Coil Pitch	The coil pitch measured in number of slots.
	Number of Strands	The number of wires per conductor (0 for auto-design).
	Wire Wrap	The thickness of the double-sided wire wrap (0 to automatically obtain this value from the wire library).
	Wire Size	The diameter of the wire (0 for auto-design). Click the button to open the **Wire Size** window where you can specify units, wire type, diameter, and gauge.
End/Insulation tab	**Input Half-turn Length**	Select or clear this check box to specify whether or not you want to enter the half-turn length. When this check box is selected, the **Half Turn Length** field appears the next time you open the **Properties** window. When this check box is selected, the **End Adjustment** field appears instead.
	Half Turn Length	The half-turn length of the armature winding.
	End Adjustment	The end length adjustment of the stator coils, which is the distance one end of the conductor extends vertically beyond the end of the stator.
	Base Inner Radius	The inner radius of the base corner.
	Tip Inner Diameter	The inner diameter of the coil tip.
	End Clearance	The end clearance between two adjacent coils.
	Slot Liner	The thickness of the slot liner insulation.
	Wedge Thickness	The thickness of the wedge insulation.
	Layer Insulation	The thickness of the insulation layer.
	Limited Fill Factor	The limited slot fill factor for the wire design. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire

		slot area.



Defining the Rotor Data for a Brushless PMDC Motor

The rotor consists of copper bars in which current is induced by the magnetic fields produced by the stator windings. In the project tree, double-click **Machine>Rotor** and **Machine-Rotor-Pole** to define the rotor and the pole.

To define general rotor data:

1. To open the **Rotor Data Properties** window, double-click the **Machine>Rotor** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the outer diameter of the rotor in the **Outer Diameter** field.
3. Enter the inner diameter of the rotor in the **Inner Diameter** field.
4. Enter the length of the rotor core in the **Length** field.
5. Select a **Steel Type** for the rotor core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
6. Enter the stacking factor for the rotor core in the **Stacking Factor** field.
7. Select a **Pole Type**:
   a. Click the button.
      The **Select Pole Type** window appears.
   b. Click a button to select the desired pole type (1, 2, 3, 4, or 5). TIP: When you run the mouse over each option, the diagram changes to show that pole type.

Note	When you place the mouse cursor over a pole type, an outline of the selected circuit type appears.

c. Click **OK** to close the **Select Pole Type** window and return to the **Properties** window.

8. Click **OK** to close the **Properties** window.

Rotor Data for Brushless PMDC Motors

To access the general rotor data, double-click the **Machine>Rotor** entry in the project tree.

The **Rotor Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the rotor core.
Inner Diameter	The inner diameter of the rotor core.
Length	The length of the rotor core.
Steel Type	The steel type of the rotor core. Click the button to open the **Select Definition** window.
Stacking Factor	The stacking factor of the rotor core.
Pole Type	The pole type for the rotor. Click this button to open the **Select Pole Type** window and select from the following types: **1**, **2**, **3**, **4**, *5*.

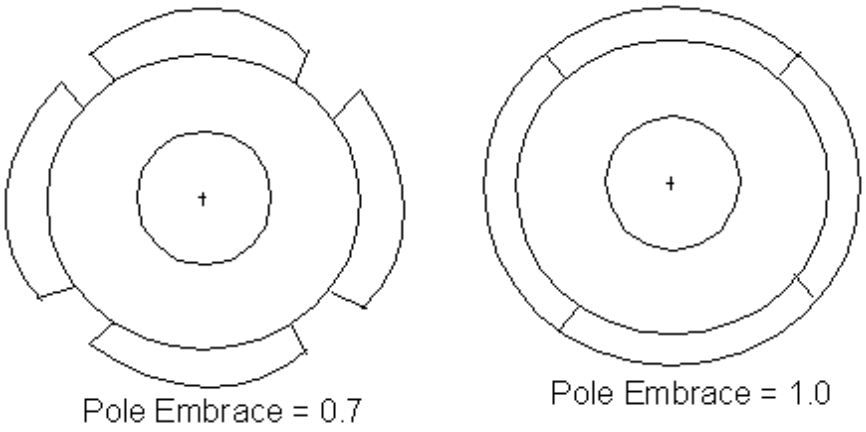
Defining the Rotor Pole for a Brushless PMDC Motor

The rotor pole drives the electromagnetic field which is coupled with the stator windings. Use the **Rotor Pole Data Properties** window to define the rotor pole.

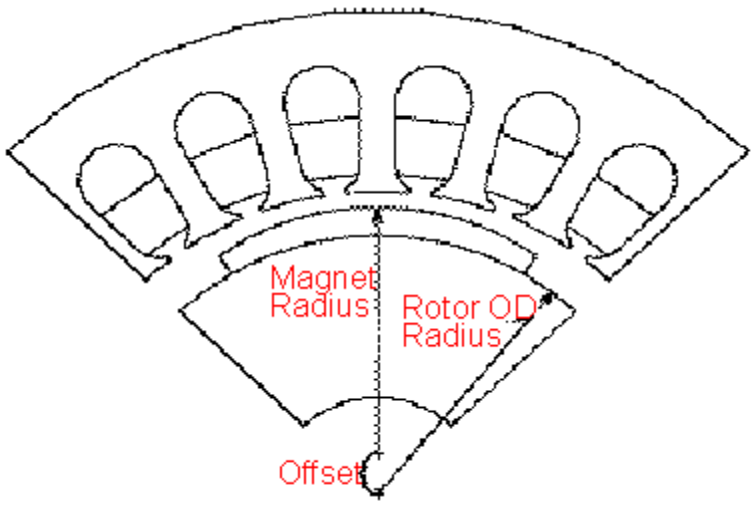
Note	Some of the fields in the **Rotor Pole** window change, or are inactive, depending on the **Rotor Type** you select.

To define the rotor pole:

1. To open the **Rotor Pole Data Properties** window, double-click the **Machine-Rotor-Pole** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. For all pole types except type **4**, enter the ratio of the actual arc distance in relation to the maximum possible arc distance in the **Embrace** field. This value is between 0 and 1.



3. For pole type **4**, enter the shaft diameter of the rotor in the **Shaft Diameter** field.
4. For pole types **1**, **2**, and **3**, enter the distance from the center of the rotor to the polar arc center in the **Offset** field. Enter 0 for a uniform air gap.



5. For pole type **5**, enter the thickness of the bridge across the two poles in the **Bridge** field.
6. For pole type **5**, enter the width of the rib supporting the bridge in the **Rib** field.
7. Select the type of magnet to use in the rotor pole from the **Magnet Type** pull-down menu.
8. For pole types **4** and **5**, enter the width of the magnet in the **Magnet Width** field.
9. Enter the maximum radial thickness of the magnet in the **Magnet Thickness** field.
10. Click **OK** to close the **Properties** window.

Rotor Pole Data for Brushless PMDC Motors

To access the pole rotor data, double-click the **Machine-Rotor-Pole** entry in the project tree.

The **Rotor Pole Data Properties** window may contain the following fields, depending on the pole type specified.

Embrace	The pole embrace. For pole types **1**, **2**, **3**, and **5**.
Shaft Diameter	The shaft diameter of the rotor. For pole type **4**.
Offset	The pole-arc center offset from the rotor center (0 for a uniform air gap). For pole types **1**, **2**, and **3**.
Bridge	The thickness of the bridge across two adjacent poles. For pole type **5**.
Rib	The width of the rib at the center of two adjacent poles that support the bridge. For pole type **5**.
Magnet Type	The type of magnet. Click the button to open the **Select Definition** window. For all pole types.
Magnet Width	The maximum width of the magnet. For pole types **4** and **5**.
Magnet Thickness	The maximum thickness of the magnet. For all pole types.

Defining the Shaft Data for a Brushless PMDC Motor

To define the shaft:

1. To open the **Shaft Data Properties** window, double-click the **Machine>Shaft** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select or clear the **Magnetic Shaft** check box to specify whether or not the shaft is to be made of magnetic material.
3. Click **OK** to close the **Properties** window.

Shaft Data for Brushless PMDC Motors

To access the shaft data, double-click the **Machine>Shaft** entry in the project tree.

The **Shaft Data Properties** window contains the following fields:

Magnetic Shaft	Select or clear this check box to indicate whether or not the shaft is made of magnetic material.

Setting Up Analysis Parameters for a Brushless PMDC Motor

To define the solution data:

1. To open the Solution Setup window, right-click **Analysis** in the project tree, and click **Add Solution Setup**.
2. Click the **General** tab. The **Operation Type** is **Motor** for this machine type.

3. Select the **Load Type** used in the motor from the following options:

Const Speed	The speed remains constant in the motor.
Const Power	The output power remains constant in the motor.
Const Torque	The torque remains constant regardless of the speed. In this case, T_{load} = T_{rated}, given by the output power divided by the given rated speed.
Linear Torque	The torque increases linearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})$ where T_{rated} is given by the output power divided by the given rated speed.
Fan Load	The load varies nonlinearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})^2$ where T_{rated} is given by the output power divided by the given rated speed.

4. Enter the output power developed at the shaft of the motor in the **Rated Output Power** field.
5. Enter the RMS line-to-line voltage in the **Rated Voltage** field.
6. Enter the desired output speed of the motor at the load point in the **Rated Speed** field.
7. Enter the temperature at which the system functions in the **Operating Temperature** field.
8. Click **OK** to close the **Solution Setup** window.

Related Topics

Solution Data for Brushless PMDC Motors

Analysis Offered

Analysis Offered

- Adapted to both Synchronous Motor and Generator

  The structures of the salient-pole synchronous motor and the generator are basically the same, but their phasor relationships and the computation methods are slightly different, their output characteristics data are also different. Therefore, RMxprt divides the synchronous machine into two design modules: Synchronous Motor and Synchronous Generator.
- Auto Arrangement of Three-phase Windings

  Almost all commonly used three-phase single- and double-layer, half- and whole-type ac windings (including fractional-pitch windings) can be automatically arranged. Users do not need to define coils one by one. RMxprt also supports a double-layer winding with half-turn coils which are auto-arranged in the order of even, odd, even, odd, …, and even, odd, as long as it is physically possible.

  When a designer adopts single-layer whole-coiled windings, RMxprt will perform winding arrangement optimization to minimize the average coil pitch. When asymmetric three-phase windings are used, winding arrangement is optimized in such a way that minimum negative-sequence and zero-sequence components are achieved.

- Winding Editor Supporting Any Single- and Double-Layer Windings

  Besides taking the great advantage of the winding auto-arrangement function in RMxprt, users can also specify any special winding by using of the Winding Editor function.

  In Winding Editor, through modification of phase belonging, number of turns, in-slot and out-slot number of each coil, it is possible to design single- and double-layer winding arrangement for any purposes.
- Analyze Air-Gap Magnetic Field Distribution

  For both uniform and non-uniform air gaps, Schwarz-Christopher Transformation is adopted to solve for the air-gap magnetic field distribution.
- Analyze EMF Waveform and Total Harmonic Distortion (THD)

  Based on the analysis of the air-gap magnetic field waveform, taking into account coil short pitch, winding distribution, skew slot, winding connection, load effects and other factors, the emf waveforms in the coils and the windings are analyzed to solve for the emf distortion factors.
- Analyze Dynamic Parameters of Damping Winding

  Different from the squirrel-cage winding of the induction machine, the damping winding of the salient-pole synchronous machine is located in the surface of magnetic field poles, which deviates greatly along the d- and the q-axes. Furthermore, the connection of damping bars has several forms. The bars under each pole could be connected, but not connected with those under other poles. All the bars could be connected together. The bars could be connected through end-plate. RMxprt can deal with all those complicated situations and give the dynamic parameters for the damping winding.

Related Topics

Setting Up Analysis Parameters for a Brushless PMDC Motor

Solution Data for Brushless PMDC Motors

To access the solution data, right-click **Analysis** in the project tree, and click **Add Solution Setup**. For this machine type, there is only one tab, the **General** tab.

The **Solution Setup** window contains the following fields:

Operation Type	The operation type is automatically set to **Motor** for this machine type.
Load Type	Select from **Const Speed**, **Const Power**, **Const Torque**, **Linear Torque**, and **Fan Load**. The default is **Const Power**.
Rated Output Power	Type a value for the rated output voltage, and select the units.
Rated Voltage	Type a value for the rated voltage, and select the units.
Rated Speed	Type a value for the rated speed, and select the units.

Operating Temperature	Type a value for the operating temperature, and select the units.

Related Topics

Setting Up Analysis Parameters for a Brushless PMDC Motor

Switched Reluctance Motors

After you have selected **Switched Reluctance Motors** as your model type, define the following:

- General data, such as the power, voltage, and speed of the motor.
- Circuit data.
- Stator core data, such as the number of poles, diameter, and yoke thickness.
- Stator coil data, such as the slot liner thickness, number of parallel branches, and number of wires in each conductor.
- Rotor core data, such as the air gap dimensions and number of poles in the rotor.
- Shaft data.
- Optional machine housing.
- Solution data.

Analysis Approach for Switched Reluctance Motors

This motor type operates with shaft position feedback to synchronize the commutation of the phase currents with precise rotor position. Typically, both the stator and the rotor are salient to increase the torque-producing characteristics of the motor. The rotor has no windings; the torque is produced by the alignment tendency of the rotor to the stator so that the stator flux linkage is maximized.

In these motors, the stator and rotor have different numbers of poles. The stator phase windings are energized at precise moments synchronized with the position of the rotor. The task of energizing the stator windings is performed by a complex electronic system.

The number of phases in the winding is the ratio of the stator number of poles to the smallest common divider of the stator and the rotor number of poles.

In switched reluctance motors (SRM), the stator and the rotor have a different number of poles, and the stator currents are commutated exactly according to rotor position. The signal of the rotor position is obtained from a position sensor. The stator windings are triggered one by one, and normally the current in a winding has finished or almost finished freewheeling when the next winding is triggered. Therefore, the mutual effects between two phases can be neglected. The voltage equation of one phase is:

$$u = u_T + R_S \cdot i + \frac{d\Psi(\theta, i)}{dt}$$

where u_T is the transistor or diode voltage drop, and R_s is the stator winding resistance. $\Psi(\theta, i)$ is the flux linkage of the winding at rotor position θ and winding current i, as is shown in Figure 8, where the rotor position when the center of the rotor slot is aligned to the winding axis is defined as 0.

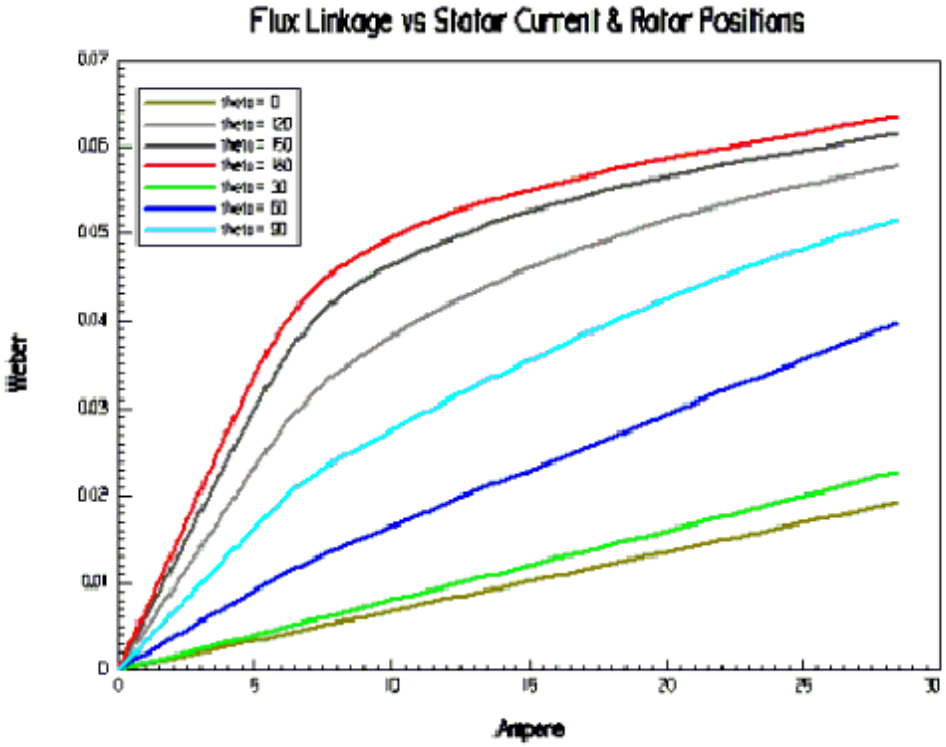


Figure 8

Let

$$L_{\theta} = \frac{\partial\Psi(\theta, i)}{\partial i}$$

and

$$G = \frac{\partial(\Psi / i)}{\partial\theta} = \frac{\partial L}{\partial\theta}$$

Then

$$u = u_T + R_S \cdot i + L\,p\,i + G\omega_e i$$

where ω_e is the rotor speed in electrical rad/s, and *p* is the differential operator as given by:

$$p = \frac{\mathbf{d}}{\mathbf{d}t}$$

The instant electromagnetic torque t_2 is:

$$t_2 = \frac{1}{2} G i^2$$

The input electric power is computed from voltage and current as:

$$P_1 = \frac{1}{T}\int_0^T (u \cdot i \cdot dt)$$

The output mechanical power is:

$$P_2 = P_1 - (P_{fw} + P_{Cua} + P_t + P_{Fe})$$

where P_{fw}, P_{Cua}, P_t, and P_{Fe} are frictional and wind loss, armature copper loss, transistor/diode loss, and iron-core loss, respectively.

The average output mechanical shaft torque T_2 is:

$$T_2 = \frac{P_2}{\omega}$$

where ω is the rotor angular speed in mechanical rad/s.

The efficiency of the electric machine is computed by:

$$\eta = \frac{P_2}{P_1} \times 100\ \%$$

Defining a Switched Reluctance Motor

The general procedure for defining a switched reluctance motor is as follows:

1. Insert a Switched Reluctance motor into a new or existing project.
2. Double-click the **Machine** entry in the project tree to define the general data.
3. Double-click the **Machine-Circuit** entry in the project tree to define the control circuit.
4. Double-click the **Machine-Stator** entry in the project tree to define the stator geometry.
5. Double-click the **Machine-Stator-Winding** entry in the project tree to define the stator windings and conductors.
6. Double-click the **Machine-Rotor** entry in the project tree to define the rotor geometry.
7. Double-click the **Machine-Shaft** entry in the project tree to define the magnetism of the shaft.
8. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
9. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
10. Choose **File>Save** to save the project.
11. Choose **RMxprt>Analyze** to analyze the design.

Note	When you place the cursor over an entry field in the data windows, a brief description of that field appears in the status bar at the bottom of the **RMxprt** window.

Once analyzed, the model can be viewed in the Maxwell 2D Modeler, or it can be used to create a new Maxwell 2D project, and a new Maxwell 3D project.

Please refer to the *Switched Reluctance Motor Problem* application note, on the technical support page of the Ansys web site, for a specific example.

Defining the General Data for a Switched Reluctance Motor

Use the **General** window to define the power settings, speed, and period of the motor.

To define the general data:

1. To open the **General Data Properties** window, double-click the **Machine** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the energy loss due to friction at the given speed in the **Frictional Loss** field.
3. Enter the wind loss due to air resistance measured at the reference speed in the **Wind Loss** field.
4. Enter the given speed in the **Reference Speed** field.
5. Select **DC** or **CCC** from the **Control Type** pull-down list.
6. Select a **Circuit Type** from the following types:

• **Full-Voltage**
• **Half-Voltage**
• **Coupled-Coil**

   The circuit types are based on industry standards. By default, type **Full-Voltage**, is selected as the circuit type.

Note	When you place the mouse cursor over a circuit type, an outline schematic of the circuit appears.

7. Click **OK** to close the **Properties** window.

General Data for Switched Reluctance Motors

To access the general data, double-click the **Machine** entry in the project tree.

The **General Data Properties** window for a three-phase induction motor contains the following fields:

Machine Type	The machine type you selected when inserting a new RMxprt design (Switched Reluctance Motor).
Frictional Loss	The frictional energy loss (due to friction) measured at the reference speed.
Wind Loss	The wind loss (due to air resistance) measured at the reference speed.
Reference Speed	The given speed of reference.

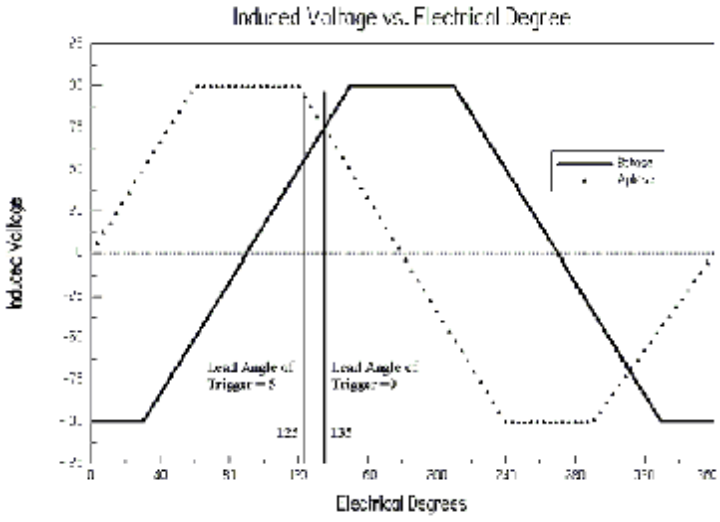
Control Type	The way the circuit is controlled. Select from **DC** or **CCC** (chopped current control, which forces the current to fall between the minimum and maximum values specified).
Circuit Type	The drive circuit type. Click the button to open the **Circuit Type** window and select from the following three types: • **Full-Voltage** • **Half-Voltage** • **Coupled-Coil**

Defining the Circuit Data for a Switched Reluctance Motor

Use the **Circuit Data Properties** window to specify the rated output power, voltage values, circuit type, and speed of the brushless DC motor.

To define the general data:

1. To open the **Circuit Data Properties** window, double-click the **Machine-Circuit** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the trigger's lead angle in electrical degrees in the **Lead Angle of Trigger** field. The trigger angle is the point at which the magnetic poles interact to begin the motion of the motor. An angle of 0 means that each phase is triggered when its axis is aligned with the rotor slot center. The trigger's lead angle is shown in the following plot of the open circuit induced voltage versus position. An angle of 0 means that the induced voltage in the triggered phase is at a maximum:



Note	A positive value represents a lead angle, and a negative value represents a lag angle.

3. Enter the period from on-status to off-status of a transistor, in electrical degrees, in the **Trigger Pulse Width** field. The trigger pulse width is the width of the energizing pulse applied to the winding, or the period for an ‘on’ status of the transistors. The maximum ‘on’ period is given by 180 degrees plus the value for the lead angle of trigger.
4. Enter the voltage drop across one transistor when the transistor is turned on in the **Transistor Drop** field. Refer to the figures of the different circuit types in step 2. This value is over one conduction path when the transistors are triggered.
5. Enter the voltage drop on all anti-parallel diodes in the discharge path in the **Diode Drop** field. If you selected a star-type circuit (**S3** or **S4**) as the **Circuit Type**, enter the total discharge voltage in this field.
6. If you selected CCC (chopped current control) as the **Control Type**, then enter the maximum and minimum current values in the **Maximum Current** and **Minimum Current** fields.
7. Click **OK** to close the **Properties** window.

Circuit Data for Switched Reluctance Motors

To access the **Circuit Data Properties** window, double-click the **Machine>Circuit** entry in the project tree. When **AC** is selected at the **Control Type**, now circuit data properties exist.

Lead Angle of Trigger	The trigger’s lead angle, in electrical degrees.
Trigger Pulse Width	The period from on-status to off-status for a transistor, in electrical degrees.
Transistor Drop	The voltage drop across one transistor when the transistor is turned on.
Diode Drop	The voltage drop across one diode in the discharge loop.
Maximum Current	The maximum current for the chopped current control. This field is not available for a DC circuit.
Minimum Current	The minimum current for the chopped current control. This field is not available for a DC circuit.

Defining the Stator Data for a Switched Reluctance Motor

The stator is the outer lamination stack where the polyphase voltage windings reside.

To define the general stator data:

1. To open the **Stator Data Properties** window, double-click the **Machine>Stator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the **Outer Diameter** of the stator.
3. Enter the **Inner Diameter** of the stator.
4. Enter the total length of the stator core in the **Length** field.

5. Enter the effective magnetic length of the core in the **Stacking Factor** field. This value typically ranges from between 0.93 and 1.0, and is defined as the total length minus the total lamination insulation, divided by the total length.
6. Select a **Steel Type** for the stator core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
7. Enter the number of poles the stator core contains in the **Number of Poles** field.
8. Enter the pole embrace in the **Embrace** field. The pole embrace is the ratio of the actual pole arc angle to the maximum possible pole angle in the field. This value ranges from between 0 and 1.
9. Enter the thickness of the stator coil yoke in the **Yoke Thickness** field.
10. Click **OK** to close the **Properties** window.

Stator Data for Switched Reluctance Motors

To access the general stator data, double-click the **Machine>Stator** entry in the project tree.

The **Stator Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the stator core.
Inner Diameter	The inner diameter of the stator core.
Length	The length of the stator core.
Stacking Factor	The stacking factor of the stator core.
Steel Type	The steel type of the stator core. Click the button to open the **Select Definition** window.
Number of Poles	The number of poles the stator core contains.
Embrace	The stator pole embrace.
Yoke Thickness	The thickness of the yoke at the stator core.

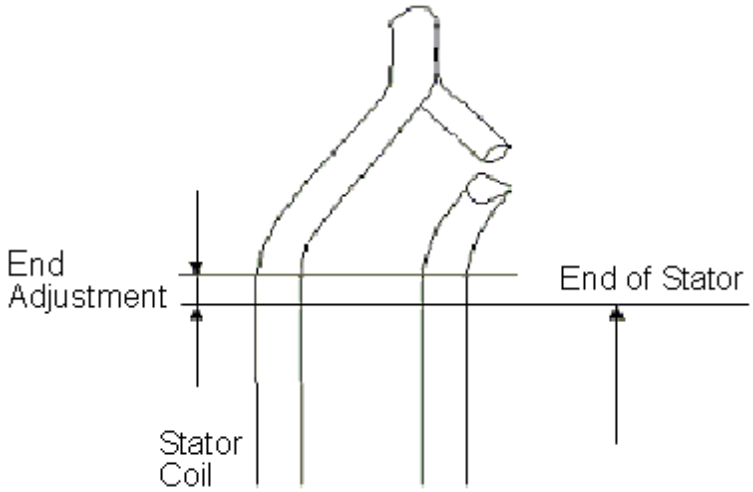
Defining the Stator Winding Data for a Switched Reluctance Motor

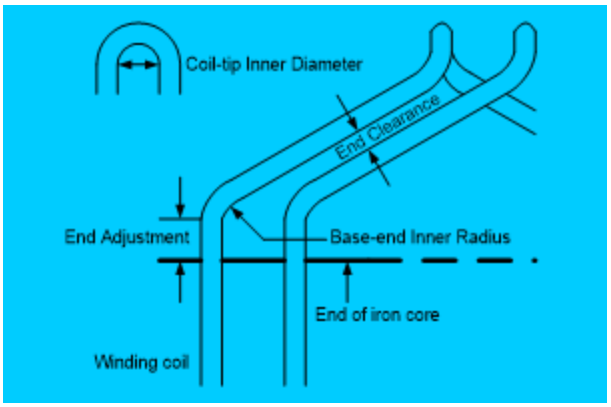
The stator coils provide the excitation for the rotating magnetic poles.

Use the **Stator Coil** window to define the parallel branches, wire specifications, and slot liner for the stator coil.

To define the stator coils:

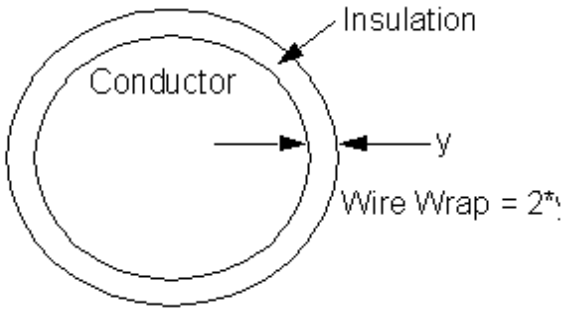
1. To open the **Stator Slot Winding Properties** window, double-click the **Machine-Stator-Winding** entry in the project tree on the desktop. You can also enter values in the **Properties** section of the desktop without opening a separate window.
2. Enter the thickness of the insulation between the stator core and the field winding in the **Insulation Thickness** field.
3. Enter the end length adjustment of the stator coils in the **End Adjustment** field. The end adjustment is the distance one end of the conductor extends vertically beyond the end of the stator.





4. Select or enter the number of parallel branches in one phase of the winding in the **Parallel Branches** field.
5. Enter the number of turns per stator pole in the **Turns per Pole** field.
6. Enter the number of wires per conductor in the **Number of Strands** field. Enter 0 to have RMxprt auto-design this value.
7. Enter the inner radius of the base corner in the **Base Inner Radius** field.

8. Enter the inner diameter of the coil tip in the **Tip Inner Diameter** field.
9. Enter the distance between two stator coils in the **End Clearance** field.
10. Enter the thickness of the slot liner insulation in the **Slot Liner** field.
11. Enter the thickness of the wedge insulation in the **Wedge Thickness** field.
12. Enter the thickness of the insulation layer in the **Layer Insulation** field.
13. Enter the limited slot fill factor for the wire design in the **Limited Fill Factor** field. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.
14. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



15. Select the **Wire Size**:
    a. Click the button for **Wire Size**.
    The **Wire Size** window appears.
    b. Select a value from the **Wire Diameter** pull-down list.
    c. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

<number>	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.
USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.
AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.
MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21mm and 2 with a diameter of 0.13mm.

The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>General Options>Machines**command.

d. When you are done setting the wire size, click **OK** to close the **Wire Size** window and return to the **Properties** window.
e. Enter the conductor area ratio of the coupled circuit to the main circuit in the **Coupled Ratio** field.
f. Click **OK** to close the **Properties** window.

Defining Different Size Wires for a Switched Reluctance Motor

Use the **Gauge** option in the **Wire Size** window if you have a conductor that is made up different size wires.

To define different size wires:

1. In the **Wire Size** window, select **MIXED** from the **Gauge** pull-down menu.
2. Select either **Round** or **Rectangular** as the **Wire Type**.
3. Enter the appropriate wire data in the table:
   - For a round wire:
     - Enter the **Diameter** in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this diameter.
   - For a rectangular wire:
     - Enter the **Width** of the wire in the table.
     - Enter the **Thickness** of the wire in the table.
     - Enter the **Fillet** value in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this data.
4. Click **Add** to add the new wire data.
5. Repeat steps 3 and 4 for each size wire you want to add.
6. When you are finished defining the wires, click **OK** to close the **Wire Size** window and return to the RMxprt **Properties** window.

Note	For example, if one conductor is made up of 5 wires, and 3 of those wires have a diameter of 0.21mm, and the other 2 have a diameter of 0.13mm, then the mixed wire size table will have two lines. The first line will list Diameter = 0.21 and Number = 3. The second line will list Diameter = 0.13 and Number = 2. An equivalent wire diameter is displayed as Wire Size value in the Winding tab in the Properties window.

Stator Winding Data for Switched Reluctance Motors

To access the stator winding data, double-click the **Machine-Stator-Winding** entry in the project tree.

The **Stator Winding Data Properties** window contains the following fields:

Insulation Thickness	The thickness of the insulation between the stator core and the field winding.
End Adjustment	The end length adjustment of the stator coils, which is the distance one end of the conductor extends vertically beyond the end of the stator.
Parallel Branches	The number of parallel branches in the stator winding.
Turns per Pole	The number of turns per stator pole (0 for auto-design).
Number of Strands	The number of wires per conductor (0 for auto-design).
Wire Wrap	The thickness of the double-sided wire wrap (0 to automatically obtain this value from the wire library).
Wire Size	The diameter of the wire (0 for auto-design). Click the button to open the **Wire Size** window where you can specify units, wire type, diameter, and gauge.
Coupled Ratio	The conductor area ratio of the coupled circuit to the main circuit.

Defining the Rotor Data for a Switched Reluctance Motor

The rotor core channels the flux generated by stator windings and provides shaft torque. The rotor consists of copper bars in which current is induced by the magnetic fields produced by the stator windings. Use the **Rotor Data Properties** window to define the air gaps, rotor dimensions, and type of steel used in the rotor core. In the project tree, double-click **Machine>Rotor** to define the rotor.

To define general rotor data:

1. To open the **Rotor Data Properties** window, double-click the **Machine>Rotor** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the outer diameter of the rotor in the **Outer Diameter** field.
3. Enter the inner diameter of the rotor in the **Inner Diameter** field.
4. Enter the length of the rotor core in the **Length** field.
5. Select a **Steel Type** for the rotor core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.

b. Select a steel type from the list, or define a new steel type.
c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.

6. Enter the effective magnetic length of the core in the **Stacking Factor** field. This value ranges from 0 to 1, and is defined as the total length minus the total lamination insulation, divided by the total length.
7. Enter the number of poles the rotor core contains in the **Number of Poles** field.
8. Enter the ratio of the actual pole angle in relation to the maximum possible pole angle in the **Embrace** field. The value ranges from 0 to 1.
9. Enter the thickness of the rotor yoke in the **Yoke Thickness** field.
10. Click **OK** to close the **Properties** window.

Rotor Data for Switched Reluctance Motors

To access the general rotor data, double-click the **Machine-Rotor** entry in the project tree.

The **Rotor Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the rotor core.
Inner Diameter	The inner diameter of the rotor core.
Length	The length of the rotor core.
Steel Type	The steel type of the rotor core. Click the button to open the **Select Definition** window.
Stacking Factor	The stacking factor of the rotor core.
Number of Poles	The number of poles the rotor core contains.
Embrace	The rotor pole embrace.
Yoke Thickness	The thickness of the rotor core yoke.

Defining the Shaft Data for a Switched Reluctance Motor

To define the shaft:

1. To open the **Shaft Data Properties** window, double-click the **Machine>Shaft** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select or clear the **Magnetic Shaft** check box to specify whether or not the shaft is to be made of magnetic material.
3. Click **OK** to close the **Properties** window.

Shaft Data for Switched Reluctance Motors

To access the shaft data, double-click the **Machine-Shaft** entry in the project tree.

The **Shaft Data Properties** window contains the following fields:

Magnetic Shaft	Select or clear this check box to indicate whether or not the shaft is made of magnetic material.

Setting Up Analysis Parameters for a Switched Reluctance Motor

To define the solution data:

1. To open the Solution Setup window, right-click **Analysis** in the project tree, and click **Add Solution Setup**.
2. Click the **General** tab. The **Operation Type** is automatically set to **Motor** for this machine type.
3. Select the **Load Type** used in the motor from the following options:

Const Speed	The speed remains constant in the motor.
Const Power	The output power remains constant in the motor.
Const Torque	The torque remains constant regardless of the speed. In this case, $T_{load} = T_{rated}$, given by the output power divided by the given rated speed.
Linear Torque	The torque increases linearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})$ where T_{rated} is given by the output power divided by the given rated speed.
Fan Load	The load varies nonlinearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})^2$ where T_{rated} is given by the output power divided by the given rated speed.

4. Enter the output power developed at the shaft of the motor in the **Rated Output Power** field.
5. Enter the RMS line-to-line voltage in the **Rated Voltage** field.
6. Enter the desired output speed of the motor at the load point in the **Rated Speed** field.
7. Enter the temperature at which the system functions in the **Operating Temperature** field.
8. Click **OK** to close the **Solution Setup** window.

Related Topics

Solution Data for Switched Reluctance Motors

Solution Data for Switched Reluctance Motors

To access the solution data, right-click **Analysis** in the project tree, and click **Add Solution Setup**. For this machine type, there is only one tab, the **General** tab.

The **Solution Setup** window contains the following fields:

Operation Type	The operation type is automatically set to **Motor** for this machine type.
Load Type	Select from **Const Speed**, **Const Power**, **Const Torque**, **Linear Torque**, and **Fan Load**. The default is **Const Power**.
Rated Output Power	Type a value for the rated output voltage, and select the units.
Rated Voltage	Type a value for the rated voltage, and select the units.
Rated Speed	Type a value for the rated speed, and select the units.
Operating Temperature	Type a value for the operating temperature, and select the units.

Related Topics

Setting Up Analysis Parameters for a Switched Reluctance Motor

Line-Start Permanent-Magnet Synchronous Motors

Once you have selected **Line-Start Permanent-Magnet Synchronous Motors** as your motor type, you can define the following:

- General data, such as the frequency, winding connection, number of poles, and voltage.
- Stator data, such as the slot type and dimensions, stator diameter, and winding data.
- Rotor pole data, such as its associated dimensions, stacking factor, and magnet type.
- Shaft data.
- Solution data.

By option, you can:

- add a machine housing.
- from a stator,
- from a rotor.

Analysis Approach for Line-Start PM Synchronous Motors

Synchronous motors use a three-phase sinusoidal voltage source to induce a rotating magnetic field in the stator. Applying this three-phase sinusoidal voltage source to the stator winding of a synchronous motor yields the rotational magnetic field in the air gap. The permanent magnet poles mounted on the rotor try to align in this rotating field, producing a synchronous torque on the rotor. Upon starting, the damping winding on the rotor generates the asynchronous starting torque, creating a self-starting feature.

The phasor diagram for the line-start permanent-magnet synchronous motor (LSSM) in the frequency domain is shown in Figure 6.

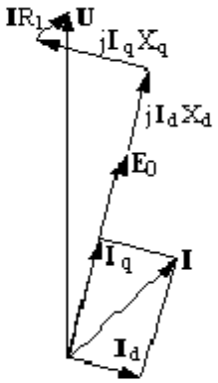


Figure 6

In Figure 6, R_1, X_d, and X_q are armature resistance, d-axis synchronous reactance, and q-axis synchronous reactance, respectively. X_d is the sum of leakage reactance, X_1 and d-axis armature reactance X_{ad}, and X_q is the sum of X_1 and q-axis armature reactance X_{aq}:

$$X_d = X_1 + X_{ad}$$

$$X_q = X_1 + X_{aq}$$

For a given torque angle θ, the angle that $\mathbf{E}_0$ lags $\mathbf{U}$, we have the following:

$$I_d X_d + I_q R_1 = U\cos\theta - E_0$$

$$-I_d R_1 + I_q X_q = U\sin\theta$$

Solving for I_d and I_q yields:

$$I_d = \frac{X_q(U\cos\theta - E_0) - R_1 U\sin\theta}{R^2_1 + X_d X_q}$$

$$I_q = \frac{R_1(U\cos\theta - E_0) - X_d U\sin\theta}{R^2_1 + X_d X_q}$$

The angle that **I** legs $\mathbf{E}_0$ is:

$$\Psi = \tanh\frac{I_d}{I_q}$$

The power factor angle (or torque angle) that **I** legs **U**, is:

$$\varphi = \Psi + \theta$$

The input power (electric power) can now be computed from voltage and current as:

$$P_1 = 3UI\cos\varphi$$

The output power (mechanical power) is:

$$P_2 = P_1 - (P_{fw} + P_{Cu} + P_{Fe})$$

where P_{fw}, P_{Cu}, and P_{Fe} are frictional and wind loss, armature copper loss, and iron-core loss, respectively.

The output mechanical power (torque) T_2 is:

$$T_2 = \frac{P_2}{\omega}$$

where ω is the synchronous speed in rad/s.

The efficiency is computed by:

$$\eta = \frac{P_2}{P_1} \times 100\,\%$$

The motor is started the same way as for an induction motor, by using a squirrel-cage-type winding -- called a damper winding in this case -- that is mounted on the rotor, producing the starting torque.

Defining a Line-Start Permanent Magnet Synchronous Motor

The general procedure for defining a line-start synchronous motor is as follows:

1. Insert a line-start synchronous motor into a new or existing project.
2. Double-click the **Machine** entry in the project tree to define the general data.
3. Double-click the **Machine-Stator** entry in the project tree to define the stator geometry.
4. Double-click the **Machine-Stator-Slot** entry in the project tree to define the stator slot dimensions.
5. Double-click the **Machine-Stator-Winding** entry in the project tree to define the stator windings and conductors.
6. Double-click the **Machine-Rotor** entry in the project tree to define the rotor geometry.
7. Double-click the **Machine-Rotor-Pole** entry in the project tree to define the pole, embrace, offset, and air gap data for the rotor pole.
8. Double-click the **Machine-Shaft** entry in the project tree to define the magnetism of the shaft.
9. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
10. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.

11. Choose **File>Save** to save the project.
12. Choose **RMxprt>Analyze** to analyze the design.

Note	When you place the cursor over an entry field in the data windows, a brief description of that field appears in the status bar at the bottom of the **RMxprt** window.

Once analyzed, the model can be viewed in the Maxwell 2D Modeler, or it can be used to create a new Maxwell 2D project, and a new Maxwell 3D design.

Defining the General Data for a Line-Start PM Synchronous Motor

Use the **General** window to define the basic parameters of the motor, such as the motor's rated output power, rated voltage, losses, and connection type.

To define the general data:

1. To open the **General Data Properties** window, double-click the **Machine** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the number of poles for the machine in the **Number of Poles** field. This value is the total number of poles in the stator (or the number of pole pairs multiplied by two).
3. Enter the energy loss due to friction at the given speed in the **Frictional Loss** field.
4. Enter the wind loss due to air resistance measured at the reference speed in the **Wind Loss** field.
5. Enter the given speed in the **Reference Speed** field.
6. Click **OK** to close the **Properties** window.

General Data for Line-Start PM Synchronous Motors

To access the general data, double-click the **Machine** entry in the project tree.

The **General Data Properties** window for a three-phase induction motor contains the following fields:

Machine Type	The machine type you selected when inserting a new RMxprt design (Line-Start PM Synchronous Motor).
Number of Poles	The number of poles the machine contains.
Frictional Loss	The frictional energy loss (due to friction) measured at the reference speed.
Wind Loss	The wind loss (due to air resistance) measured at the reference speed.
Reference Speed	The given speed of reference.

Defining the Stator Data for a Line-Start PM Synchronous Motor

The stator is the outer lamination stack where the polyphase voltage windings reside.

Use the **Stator Data**, **Stator Slot Data**, and **Stator Winding Data** windows to define the stator data, such as physical dimensions of the lamination, windings, and conductors.

To define the general stator data:

1. To open the **Stator Data Properties** window, double-click the **Machine>Stator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the **Outer Diameter** of the stator.
3. Enter the **Inner Diameter** of the stator.
4. Enter the length of the stator core in the **Length** field.
5. Enter the effective magnetic length of the core in the **Stacking Factor** field.
6. Select a **Steel Type** for the stator core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
7. Enter the **Number of Slots** in the stator.
8. Select the **Slot Type**:
   a. Click the button for the **Slot Type**.
      The **Select Slot Type** window appears.
   b. Select a slot type (available types include 1 through 4).

Note	When you place the mouse cursor over the slot type, a schematic of the selected type appears, displaying the slot dimension variables.

   c. Click **OK** to close the **Select Slot Type** window and return to the **Properties** window.
9. Enter the skew width, measured in slot number, in the **Skew Width** field.
10. Click **OK** to close the **Properties** window.

Stator Data for Line-Start PM Synchronous Motors

To access the general stator data, double-click the **Machine>Stator** entry in the project tree.

The **Stator Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the stator core.
Inner Diameter	The inner diameter of the stator core.
Length	The length of the stator core.
Stacking Factor	The stacking factor of the stator core.

Steel Type	The steel type of the stator core. Click the button to open the **Select Definition** window.
Number of Slots	The number of slots the stator core contains.
Slot Type	The type of slots in the stator core. Click the button to open the **Select Slot Type** window.
Skew Width	The skew width measured in slot number.

Defining the Stator Slots for a Line-Start PM Synchronous Motor

To define the slot type:

1. To open the **Stator Slot Data Properties** window, double-click the **Machine-Stator-Slot** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Optionally, to automatically design the dimensions of slots **Hs2**, **Bs1**, and **Bs2**, select the **Auto Design** check box.
3. Optionally, to design dimensions of slots **Bs1** and **Bs2** based on the stator tooth width, select the **Parallel Tooth** check box, and enter a value in the **Tooth Width** field.
4. Enter the available slot dimensions.

Hs0	Always available.
Hs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.
Bs0	Always available.
Bs1	Available only when **Auto Design** and **Parallel Tooth** are both cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Bs2	Available only when **Auto Design** and **Parallel Tooth** are both cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Rs	Rs is added when the slot type is 3 or 4.

5. Click **OK** to close the **Properties** window.

Stator Slot Data for Line-Start PM Synchronous Motors

To access the stator slot data, double-click the **Machine-Stator-Slot** entry in the project tree.

The **Stator Slot Data Properties** window contains the following fields:

Auto Design	Select or clear this to enable or disable auto-design of slots **Hs2**, **Bs1**, and **Bs2**. When this check box is selected, only two other fields appear in the window: **Hs0**

	and **Bs0**.
Parallel Tooth	Select this to design **Bs1** and **Bs2** based on the tooth width. When this check box is selected, the **Bs1** and **Bs2** fields are removed, and the **Tooth Width** field is added.
Tooth Width	The tooth width for the parallel tooth, on which **Bs1** and **Bs2** are designed.
Hs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Hs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Hs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Rs	A slot dimension. (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected). Rs is added when the slot type is 3 or 4.

Defining the Stator Windings and Conductors for a Line-Start PM Synchronous Motor

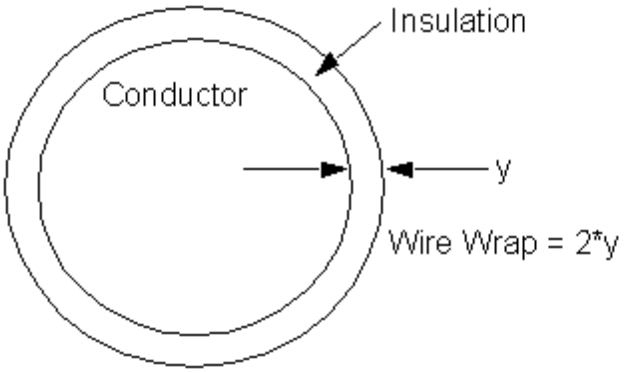
To define the stator windings and conductors:

1. To open the **Stator Slot Winding Properties** window, double-click the **Machine-Stator-Winding** entry in the project tree on the desktop. You can also enter values in the **Properties** section of the desktop without opening a separate window.
2. Click the **Winding** tab.
3. Enter the number of layers in the stator winding in the **Winding Layers** field.
4. Select the **Winding Type** for the stator:
   a. Click the button for **Winding Type**.
   The **Winding Type** window appears.
   b. Select from one of the following three types of winding:
      - **Whole Coiled**
      - **Half Coiled**
      - **Editor**

Note	When you place the mouse cursor over the winding type, a schematic of that type appears.

   c. Click **OK** to close the **Winding Type** window and return to the **Properties** window.
5. Select or enter the number of parallel branches in one phase of the winding in the **Parallel Branches** field.

6. Enter the total number of conductors in each stator slot in the **Conductors per Slot** field. This value is the number of turns per coil multiplied by the number of layers.
7. Enter the coil pitch, measured in number of slots, in the **Coil Pitch** field. The coil pitch is the number of slots separating one winding. For example, if a coil starts in slot 1 and ends in slot 6, it has a coil pitch of 5.
8. Enter the number of wires per conductor in the **Number of Strands** field. Enter 0 to have RMxprt auto-design this value.
9. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



10. Select the **Wire Size**:
    a. Click the button for **Wire Size**.
    The **Wire Size** window appears.
    b. Select a value from the **Wire Diameter** pull-down list.
    c. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

< ***number*** **>**	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.
USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.
AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.
MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21mm and 2 with a diameter of 0.13mm.

The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>General Options>Machines**command.

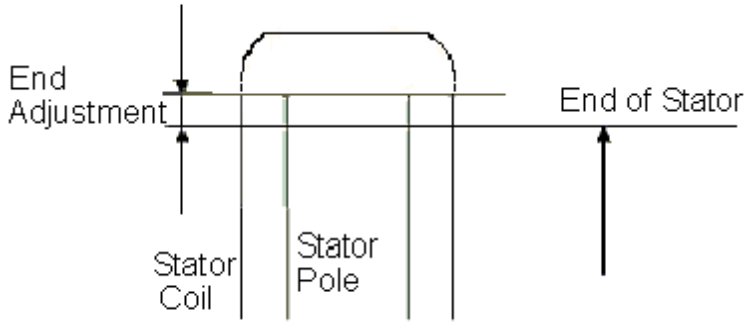
d. When you are done setting the wire size, click **OK** to close the **Wire Size** window and return to the **Properties** window.

e. Click the **End/Insulation** tab.

f. Select or clear the **Input Half-turn Length** check box.

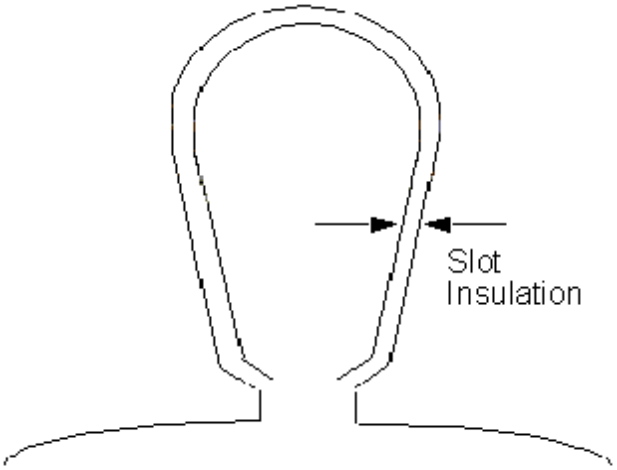
g. Do one of the following:

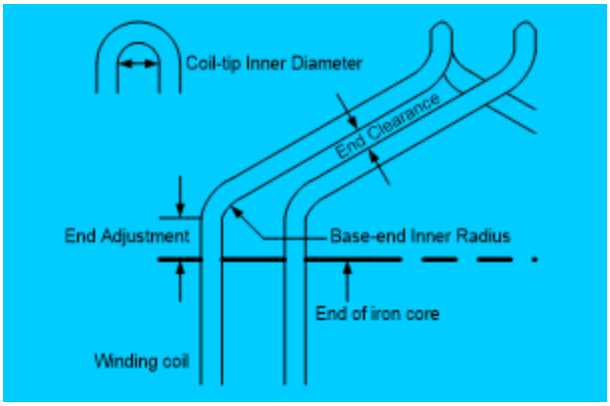
- If you selected **Input Half-turn Length**, then enter the half-turn length of the armature winding in the Half Turn Length field.
- If you cleared **Input Half-turn Length**, then enter the end length adjustment of the stator coils in the **End Adjustment** field. The end adjustment is the distance one end of the conductor extends vertically beyond the end of the stator.



11. Enter the inner radius of the base corner in the **Base Inner Radius** field.
12. Enter the inner diameter of the coil tip in the **Tip Inner Diameter** field.
13. Enter the distance between two stator coils in the **End Clearance** field.

14. Enter the thickness of the slot liner insulation in the **Slot Liner** field.





15. Enter the thickness of the wedge insulation in the **Wedge Thickness** field.
16. Enter the thickness of the insulation layer in the **Layer Insulation** field.
17. Enter the limited slot fill factor for the wire design in the **Limited Fill Factor** field. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.
18. Click **OK** to close the **Properties** window.

Winding Editor for a Line-Start Synchronous Motor

For a line-start synchronous motor, you may want to specify a different number of conductors for each stator slot. The Winding Editor makes this possible by enabling you to specify the number of turns for each coil.

To specify the number of turns for each coil:

1. Click **Machine>Winding>Edit Layout**.
   The **Winding Editor** window appears.
2. In the table in the upper left, set which phase you want for each coil and which slot is the “in” and “out” slot for the current in each coil.
3. If you are working on a quarter or half model, you may want to specify a multiplier by clicking the **Periodic Multiplier** check box and specifying a value.
4. Select or deselect the **Constant Turns** or **Constant Pitch** check boxes, depending on whether you want to be able to change these setting in the table above. When these options are selected, you cannot change the turns or pitch.
5. When you are satisfied with the coil settings, click **OK** to close the **Winding Editor** window.

Defining Different Size Wires for a Line-Start Synchronous Motor

Use the **Gauge** option in the **Wire Size** window if you have a conductor that is made up different size wires.

To define different size wires:

1. In the **Wire Size** window, select **MIXED** from the **Gauge** pull-down menu.
2. Select either **Round** or **Rectangular** as the **Wire Type**.
3. Enter the appropriate wire data in the table:
   - For a round wire:
     - Enter the **Diameter** in the table.
     - Enter a **Number** in the table to specify how many of the conductor’s wires have this diameter.
   - For a rectangular wire:
     - Enter the **Width** of the wire in the table.
     - Enter the **Thickness** of the wire in the table.
     - Enter the **Fillet** value in the table.
     - Enter a **Number** in the table to specify how many of the conductor’s wires have this data.
4. Click **Add** to add the new wire data.
5. Repeat steps 3 and 4 for each size wire you want to add.
6. When you are finished defining the wires, click **OK** to close the **Wire Size** window and return to the RMxprt **Properties** window.

Note	For example, if one conductor is made up of 5 wires, and 3 of those wires have a diameter of 0.21mm, and the other 2 have a diameter of 0.13mm, then the mixed wire size table will have two lines. The first line will list Diameter = 0.21 and Number = 3. The second line will list Diameter = 0.13 and Number = 2. An equivalent wire diameter is displayed as Wire Size value in the Winding tab in the Properties window.

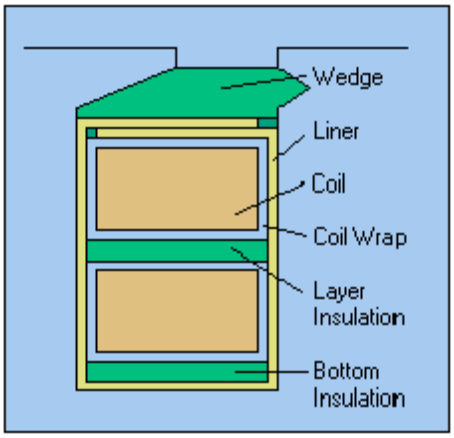
Stator Winding Data for Line-Start PM Synchronous Motors

To access the stator winding data, double-click the **Machine-Stator-Winding** entry in the project tree.

The **Stator Winding Data Properties** window contains the following fields:

Winding tab	**Winding Layers**	The number of winding layers.
	Winding Type	The type of stator winding. Click the button to open the **Winding Type** window and choose from **Whole Coiled**, **Half Coiled**, and **Editor**.
	Parallel Branches	The number of parallel branches in the stator winding.
	Conductors per Slot	The number of conductors per stator slot (0 for auto-design).
	Coil Pitch	The coil pitch measured in number of slots.
	Number of Strands	The number of wires per conductor (0 for auto-design).
	Wire Wrap	The thickness of the double-sided wire wrap (0 to automatically obtain this value from the wire library).
	Wire Size	The diameter of the wire (0 for auto-design). Click the button to open the **Wire Size** window where you can specify units, wire type, diameter, and gauge.
End/Insulation tab	**Input Half-turn Length**	Select or clear this check box to specify whether or not you want to enter the half-turn length. When this check box is selected, the **Half Turn Length** field appears the next time you open the **Properties** window. When this check box is selected, the **End Adjustment** field appears instead.
	Half Turn Length	The half-turn length of the armature winding.
	End Adjustment	The end length adjustment of the stator coils, which is the distance one end of the conductor extends vertically beyond the end of the stator.
	Base Inner Radius	The inner radius of the base corner.

	Tip Inner Diameter	The inner diameter of the coil tip.
	End Clearance	The end clearance between two adjacent coils.
	Slot Liner	The thickness of the slot liner insulation.
	Wedge Thickness	The thickness of the wedge insulation.
	Layer Insulation	The thickness of the insulation layer.
	Limited Fill Factor	The limited slot fill factor for the wire design. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.



Optional Vent for Line-Start PM Synchronous Motor Stator

To add a Vent to the stator, select the stator icon and right-click to display the pop-up menu with **Insert Vent**.

The vent is shown in the project tree under the stator.

To remove an existing Vent, select the stator and right-click to display the up-up menu with **Remove Vent**.

The Vent Data properties window contains the following fields.

Vent Ducts	Number of radial vent ducts
Duct Width	Width of radial vent ducts
Magnetic spacer width	Width of magnetic spacer which hold vent ducts. 0 for non-magnetic spacer.
Duct pitch	Vent ducts.

Defining the Rotor Data for a Line-Start PM Synchronous Motor

The rotor consists of copper bars in which current is induced by the magnetic fields produced by the stator windings. In the project tree, double-click **Machine-Rotor** and **Machine-Rotor-Pole** to define the rotor and the pole.

To define general rotor data:

1. To open the **Rotor Data Properties** window, double-click the **Machine>Rotor** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the outer diameter of the rotor in the **Outer Diameter** field.
3. Enter the inner diameter of the rotor in the **Inner Diameter** field.
4. Enter the length of the rotor core in the **Length** field.
5. Select a **Steel Type** for the rotor core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
6. Enter the effective magnetic length of the rotor core in the **Stacking Factor** field. This value ranges from 0 to 1 and is defined as the total length minus the total lamination insulation, divided by the total length. A value of 1 indicates that the rotor is not laminated.
7. Select a **Pole Type**:
   a. Click the button.
      The **Select Pole Type** window appears.
   b. Click a button to select the desired pole type (1, 2, 3, 4, 5, 6, 7, or 8). TIP: When you run the mouse over each option, the diagram changes to show that pole type.

Note	When you place the mouse cursor over a pole type, an outline of the selected circuit type appears.

   c. Click **OK** to close the **Select Pole Type** window and return to the **Properties** window.
8. Click **OK** to close the **Properties** window.

Rotor Data for Line-Start PM Synchronous Motors

To access the general rotor data, double-click the **Machine>Rotor** entry in the project tree.

The **Rotor Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the rotor core.
Inner Diameter	The inner diameter of the rotor core.
Length	The length of the rotor core.

Steel Type	The steel type of the rotor core. Click the button to open the **Select Definition** window.
Stacking Factor	The stacking factor of the rotor core.
Pole Type	The pole type for the rotor. Click this button to open the **Select Pole Type** window and select from the following types: **1**, **2**, **3**, **4**, **5**, **6**, **7**, **8**. When you mouse over each button, a diagram appears for that pole type, showing the arrangement and dimensions.

Defining the Rotor Pole for a Line-Start PM Synchronous Motor

The rotor pole drives the electromagnetic field which is coupled with the stator windings. Use the **Rotor Pole Data Properties** window to define the rotor pole.

Note	Some of the fields in the **Rotor Pole** window change, or are inactive, depending on the **Rotor Type** you select.

To define the rotor pole:

1. To open the **Rotor Pole Data Properties** window, double-click the **Machine-Rotor-Pole** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the limited diameter for the magnet ducts in the **D1** field.
3. Enter one or more of the following magnet duct dimensions, depending on the pole type selected: **O1**, **O2**, **B1**.
4. For all pole types except number **8**, enter the width of the rib supporting the bridge in the **Rib** field.
5. Select the type of magnet to use in the rotor pole:
   a. Click **Magnet Type** button.
      The **Select Definition** window appears.
   b. Select a material.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
6. Enter the total width of all magnets per pole in the **Magnet Width** field.
7. Enter the maximum radial thickness of the magnet in the **Magnet Thickness** field.
8. Click **OK** to close the **Properties** window.

Rotor Pole Data for Line-Start PM Synchronous Motors

To access the pole rotor data, double-click the **Machine-Rotor-Pole** entry in the project tree.

The **Rotor Pole Data Properties** window contains the following fields:

D1	The limited diameter for the magnet ducts. See the diagrams in the **Select Pole Type** window for the location of each dimension and which pole types require which dimensions.

O1	A magnet duct dimension. See the diagrams in the **Select Pole Type** window for the location of each dimension and which pole types require which dimensions.
O2	A magnet duct dimension. See the diagrams in the **Select Pole Type** window for the location of each dimension and which pole types require which dimensions.
B1	A magnet duct dimension. See the diagrams in the **Select Pole Type** window for the location of each dimension and which pole types require which dimensions.
Rib	The width of the rib at the center of two adjacent poles that support the bridge. For pole types except number **8**.
Magnet Type	The type of magnet. Click the button to open the **Select Definition** window. For all pole types.
Magnet Width	The maximum width of the magnet. For all pole types.
Magnet Thickness	The maximum thickness of the magnet. For all pole types.

Optional Rotor Damper for Line-Start PM Synchronous Motor

- To add a damper, right-click on the rotor item in the project tree to display the pop-up menu with **Insert Damper**.
- To remove an existing damper, right-click on the rotor icon in the project tree to display the shortcut menu with **Remove Damper**.

The **Damper Data** properties window contains the following fields.

Damper Slots per Pole	Number of damper slots per pole.
Slot Type	Damper slot type. Click the field button open the Slot selection window and select one of the four types.
Cast Rotor.	Specify whether the rotor squirrel cage winding is cast.
Bar conductor type	Click the field button to open the Materials Selection window to specify the material for the bar conductor.
End Length	Single side end extended bar length
End Ring Width	Axial width of end ring.
End Ring Height	Radial height of end ring
End Ring Conductor type	Click the field button to open the Materials Selection window to specify the material for the end ring conductor.

Defining the Shaft Data for a Line-Start PM Synchronous Motor

To define the shaft:

1. To open the **Shaft Data Properties** window, double-click the **Machine-Shaft** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select or clear the **Magnetic Shaft** check box to specify whether or not the shaft is to be made of magnetic material.
3. Click **OK** to close the **Properties** window.

Shaft Data for Line-Start PM Synchronous Motors

To access the shaft data, double-click the **Machine-Shaft** entry in the project tree.

The **Shaft Data Properties** window contains the following fields:

Magnetic Shaft	Select or clear this check box to indicate whether or not the shaft is made of magnetic material.

Setting Up Analysis Parameters for a Line-Start PM Synchronous Motor

To define the solution data:

1. To open the Solution Setup window, right-click **Analysis** in the project tree, and click **Add Solution Setup**.
2. Click the **General** tab. The **Operation Type** is automatically set to **Motor** for this machine type.
3. Select the **Load Type** used in the motor from the following options:

Const Speed	The speed remains constant in the motor.
Const Power	The output power remains constant in the motor.
Const Torque	The torque remains constant regardless of the speed. In this case, $T_{load} = T_{rated}$, given by the output power divided by the given rated speed.
Linear Torque	The torque increases linearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})$ where T_{rated} is given by the output power divided by the given rated speed.
Fan Load	The load varies nonlinearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})^2$ where T_{rated} is given by the output power divided by the given rated speed.

4. Enter the output power developed at the shaft of the motor in the **Rated Output Power** field.
5. Enter the RMS line-to-line voltage in the **Rated Voltage** field.
6. Enter the desired output speed of the motor at the load point in the **Rated Speed** field.
7. Enter the temperature at which the system functions in the **Operating Temperature** field.
8. Click the **Line-Start PM Synchronous Motor** tab.

9. Select **Wye** or **Delta** from the **Winding Connection** pull-down list.
10. Click **OK** to close the **Solution Setup** window.

Related Topics

Solution Data for Line-Start PM Synchronous Motors

Solution Data for Line-Start PM Synchronous Motors

To access the solution data, right-click **Analysis** in the project tree, and click **Add Solution Setup**. For this machine type, there is only one tab, the **General** tab.

The **Solution Setup** window contains the following fields:

Operation Type	***General** tab*. The operation type is automatically set to **Motor** for this machine type.
Load Type	***General** tab*. Select from **Const Speed**, **Const Power**, **Const Torque**, **Linear Torque**, and **Fan Load**. The default is **Const Power**.
Rated Output Power	***General** tab*. Type a value for the rated output voltage, and select the units.
Rated Voltage	***General** tab*. Type a value for the rated voltage, and select the units.
Rated Speed	***General** tab*. Type a value for the rated speed, and select the units.
Operating Temperature	***General** tab*. Type a value for the operating temperature, and select the units.
Winding Connection	***Line-Start PM Synchronous Motor** tab.* Select **Wye** or **Delta** from the **Winding Connection** pull-down list.

Related Topics

Setting Up Analysis Parameters for a Line-Start PM Synchronous Motor

Universal Motors

After you have selected **Universal Motors** as your model type, enter the motor data to define the following:

- General data, such as the number of poles, frictional loss, and reference speed.
- Stator pole and winding data, such as its associated pole dimensions, type of steel, and wire definitions.
- Rotor data, such as the slot types and dimensions, rotor diameter, laminations, and windings and conductors.
- Commutator and brush data, such as the commutator dimensions and brush length.
- Shaft data.

- Optional machine housing.
- Solution data.

Analysis Approach for Universal Motors

For a DC motor, if its field winding is connected in series with its armature winding, it becomes a series motor. When the polarity of the terminal voltage changes, the direction of the produced electromagnetic torque does not change because the armature and the exciting currents alternate their directions at the same time. That means the motor can operate not only with a DC source but also with an AC source. Because it can operate with both DC and AC sources, a series motor is also called universal motor (UniM).

For a universal motor, the stator is equipped with *p* pairs of coil-wound poles, creating *P* pairs of alternating north and south poles. The coil excitation may be either AC or DC. The rotor is equipped with a distributed winding connected to a commutator that revolves together with the rotor.

A system of brushes is kept in permanent electrical contact with the commutator. When AC or DC current is applied to the rotor winding (via the brushes and commutator) a torque is produced by the interaction of the rotor (armature) currents and the field produced by the stator poles.

The commutator causes the armature to create a magnetic flux distribution whose axis is perpendicular to the axis of the field flux produced by the permanent magnets. For these motors, the commutator acts as a mechanical rectifier.

The performance of a universal motor is analyzed in the frequency domain. The voltage equation of a universal motor is:

$$U = ZI = (R_a + R_f + R_b)I + j\omega(L_a + L_f + 2M_{af})I + \omega_e(G_{aa} + G_{af})I$$

where, R_a, R_f, and R_b are the armature resistance, field winding resistance, and the brush contact resistance, respectively. L_a, L_f, and M_{af} are the armature self inductance, field winding self inductance, and their mutual inductance, respectively, and are linearized nonlinear parameters. Gaa and Gaf are the coefficients of motion induced voltages by the armature and field winding currents, respectively, and are also linearized nonlinear parameters. ω is the radian frequency, and ω_e the rotor speed in electric rad/s. Z is equivalent input impedance. When the brush axis is aligned with q-axis:

$$M_{af} = G_{aa} = 0$$

For a given rotor speed ω_e, armature current can be computed based on the applied voltage *U*, as:

$$I = \frac{U}{Z}$$

The input power (electric power) is directly computed from voltage and current as:

$$P_1 = UI\cos\varphi$$

The output power (mechanical power) is:

$$P_2 = P_1 - (P_{fw} + P_b + P_{cuf} + P_{Fe})$$

where P_{fw}, P_b, P_{cua}, P_{cuf}, and P_{Fe} are frictional and wind loss, brush drop loss, armature copper loss, field winding copper loss, and iron-core loss, respectively.

The output mechanical shaft torque T_2 is:

$$T_2 = \frac{P_2}{\omega}$$

The efficiency is computed by:

$$eff = \frac{P_2}{P_1} \times 100\ \%$$

Defining a Universal Motor

The general procedure for defining a universal motor is as follows:

1. Insert a universal motor into a new or existing project.
2. Double-click the **Machine** entry in the project tree to define the general data.
3. Double-click the **Machine-Stator** entry in the project tree to define the stator geometry.
4. Double-click the **Machine-Stator-Pole** entry in the project tree to define the stator pole dimensions.
5. Double-click the **Machine-Stator-Winding** entry in the project tree to define the stator windings and conductors.
6. Double-click the **Machine-Rotor** entry in the project tree to define the rotor geometry.
7. Double-click the **Machine-Rotor-Slot** entry in the project tree to define the rotor slot dimensions.
8. Double-click the **Machine-Rotor-Winding** entry in the project tree to define the rotor windings and conductors.
9. Double-click the **Machine-Commutator** entry in the project tree to define the commutator and brush data.
10. Double-click the **Machine-Shaft** entry in the project tree to define the magnetism of the shaft.
11. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
12. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
13. Choose **File>Save** to save the project.
14. Choose **RMxprt>Analyze** to analyze the design.

Note	When you place the cursor over an entry field in the data windows, a brief description of that field appears in the status bar at the bottom of the **RMxprt** window.

Once analyzed, the model can be viewed in the Maxwell 2D Modeler, or it can be used to create a new Maxwell 2D project, and a new Maxwell 3D design.

Refer to the *Universal Motor Problem* application note, on the technical support page of the Ansys web site, for a specific example.

Defining the General Data for a Universal Motor

Use the **General** window to define the basic parameters of the universal motor such as the power settings, speed, and rated voltage.

To define the general data:

1. To open the **General Data Properties** window, double-click the **Machine** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the number of poles for the machine in the **Number of Poles** field. This value is the total number of poles in the stator (or the number of pole pairs multiplied by two).
3. Enter the energy loss due to friction at the given speed in the **Frictional Loss** field.
4. Enter the wind loss due to air resistance measured at the reference speed in the **Wind Loss** field.
5. Enter the given speed in the **Reference Speed** field.
6. Click **OK** to close the **Properties** window.

General Data for Universal Motors

To access the general data, double-click the **Machine** entry in the project tree.

The **General Data Properties** window for a three-phase induction motor contains the following fields:

Machine Type	The machine type you selected when inserting a new RMxprt design (Switched Reluctance Motor).
Number of Poles	Number of poles for this machine.
Frictional Loss	The frictional energy loss (due to friction) measured at the reference speed.
Wind Loss	The wind loss (due to air resistance) measured at the reference speed.
Reference Speed	The given speed of reference.

Defining the Stator Data for a Universal Motor

Use the **Stator Properties** windows to define the stator dimensions, slots, windings, and conductors.

The stator is the outer lamination stack where the polyphase voltage windings reside.

To define the general stator data:

1. To open the **Stator Data Properties** window, double-click the **Machine>Stator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the **Outer Diameter** of the stator.
3. Enter the overall width of the stator outer profile in the **Overall Width** field.
4. Enter the **Inner Diameter** of the stator.
5. Enter the length of the stator core in the **Length** field.
6. Enter the stacking factor for the stator core in the **Stacking Factor** field.
7. Select a **Steel Type** for the stator core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
8. Click **OK** to close the **Properties** window.

Stator Data for Universal Motors

To access the general stator data, double-click the **Machine>Stator** entry in the project tree.

The **Stator Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the stator core.
Overall Width	The overall width of the stator outer profile.
Inner Diameter	The inner diameter of the stator core.
Length	The length of the stator core.
Stacking Factor	The effective magnetic length of the stator core.
Steel Type	The steel type of the stator core. Click the button to open the **Select Definition** window.

Defining the Stator Pole for a Universal Motor

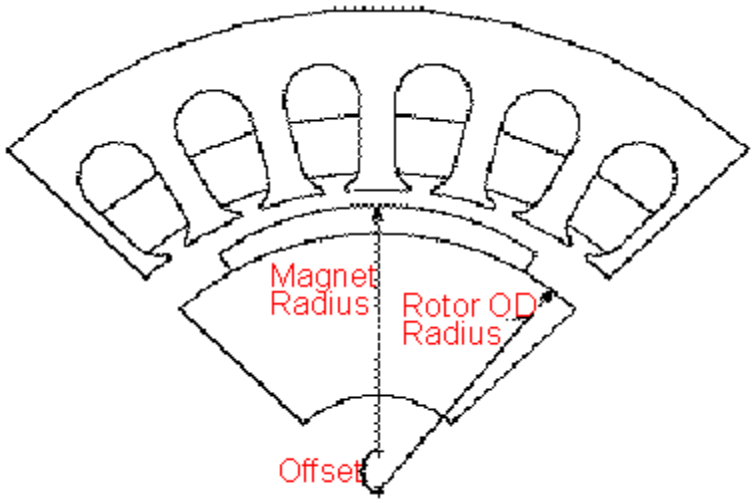
The rotor pole drives the electromagnetic field which is coupled with the stator windings. Use the **s Stator Pole Data Properties** window to define the stator pole.

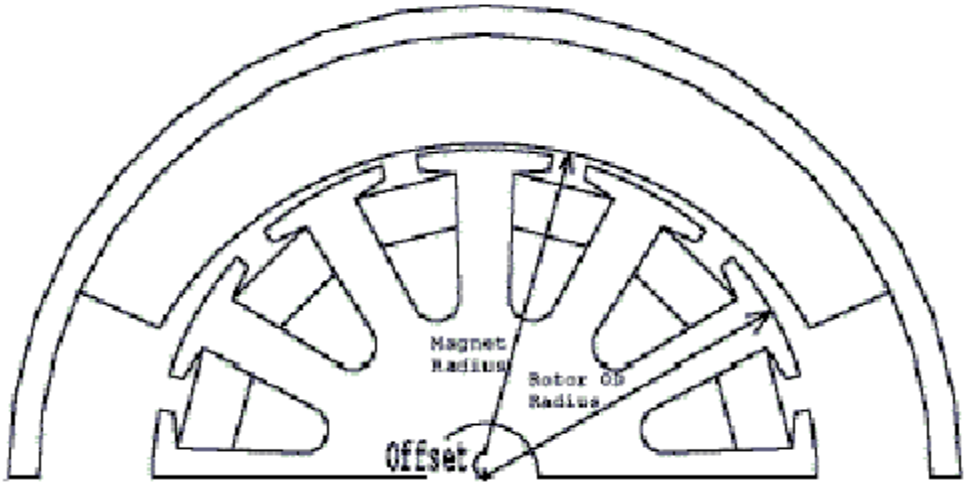
To define the rotor pole:

1. To open the **Stator Pole Data Properties** window, double-click the **Machine-Stator-Pole** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)

Note	For a two-pole machine, a pole embrace of 0.75 yields a magnet with a span of 135 degrees (based on 0.75*180 degrees).

2. Enter the ratio of the actual arc distance in relation to the maximum possible arc distance in the **Embrace** field. This value is between 0 and 1.
3. Enter the distance from the center of the stator to the magnet arc center in the **Offset** field. Enter 0 for a uniform air gap.





4. Enter the minimum pole width in the **PoleWidth** field.
5. Enter the yoke thickness in the **Ty** field.

6. Enter the shoe-tip thickness in the **Ts** field.
7. Enter the pole's hole radius in the **R1** field. If there is no hole in the design, enter **0**.
8. Enter the pole's side fillet radius in the **R2** field.
9. Enter the radius of the pole's center side fillet arcs in the **R3** field.
10. Enter the radius of the shoe connecting arc in the **R4** field. To auto-design this dimension, enter **0**. For a linear connection, enter **0**.
11. Enter the inner radius of the screw hole between the two poles in the **R5** field. If there is no hole in the design, enter **0**.
12. Enter the outer radius of the screw hole between the two poles in the **R6** field. If there is no hole in the design, enter **0**.
13. Click **OK** to close the **Properties** window.

Stator Pole Data for Universal Motors

To access the stator pole data, double-click the **Machine-Stator-Pole** entry in the project tree.

The **Stator Pole Data Properties** window contains the following fields:

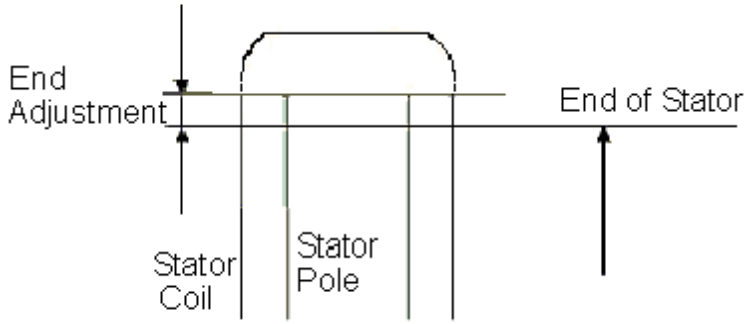
Embrace	The pole embrace.
Offset	The pole-arc center offset from the stator center (0 for a uniform air gap).
PoleWidth	The minimum pole width.
Ty	The yoke thickness.
Ts	The shoe-tip thickness.
R1	The hole radius in the pole (0 for no hole).
R2	The radius of the pole side fillet.
R3	The radius of the center of the pole side fillet arcs.
R4	The radius of the shoe connecting arc (0 for auto-design or for a linear connection).
R5	The inner radius of the screw hole between two poles (0 for no hole).
R6	The outer radius of the screw hole between two poles (0 for no hole).

Defining the Stator Windings and Conductors for a Universal Motor

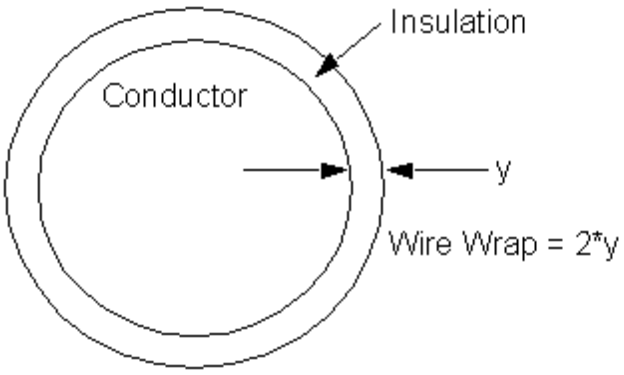
To define the stator windings and conductors:

1. To open the **Stator Winding Properties** window, double-click the **Machine-Stator-Winding** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the thickness of the insulation between the stator core and the field winding in the **Insulation Thickness** field.
3. Enter the end length adjustment of the stator coils in the **End Adjustment** field. The end adjustment is the distance one end of the conductor extends vertically beyond the end of

the stator.



4. Enter the number of parallel branches in the stator winding in the **Parallel Branches** field.
5. Enter the number of turns per stator pole in the **Turns per Pole** field. To auto-design the number of turns, enter **0**.
6. Enter the number of wires per conductor in the **Number of Strands** field. Enter **0** to have RMxprt auto-design this value.
7. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



8. Select the **Wire Size**:
   a. Click the button for **Wire Size**.
      The **Wire Size** window appears.
   b. Select a value from the **Wire Diameter** pull-down list.
   c. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

< ***number*** **>**	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.

USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.
AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.
MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21mm and 2 with a diameter of 0.13mm.

The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>General Options>Machines**command.

a. When you are done setting the wire size, click **OK** to close the **Wire Size** window and return to the **Properties** window.

b. Enter the thickness of the wedge insulation in the **Wedge Thickness** field.

c. Enter the thickness of the insulation layer in the **Layer Insulation** field.

d. Enter the limited slot fill factor for the wire design in the **Limited Fill Factor** field. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.

e. Click **OK** to close the **Properties** window.

Defining Different Size Wires for a Universal Motor Stator Winding

To define different size wires:

1. In the **Wire Size** window, select **MIXED** from the **Gauge** pull-down menu.
2. Select either **Round** or **Rectangular** as the **Wire Type**.
3. Enter the appropriate wire data in the table:
   - For a round wire:
     - Enter the **Diameter** in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this diameter.
   - For a rectangular wire:
     - Enter the **Width** of the wire in the table.
     - Enter the **Thickness** of the wire in the table.
     - Enter the **Fillet** value in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this data.
4. Click **Add** to add the new wire data.
5. Repeat steps 3 and 4 for each size wire you want to add.

6. When you are finished defining the wires, click **OK** to close the **Wire Size** window and return to the RMxprt **Properties** window.

Note	For example, if one conductor is made up of 5 wires, and 3 of those wires have a diameter of 0.21mm, and the other 2 have a diameter of 0.13mm, then the mixed wire size table will have two lines. The first line will list Diameter = 0.21 and Number = 3. The second line will list Diameter = 0.13 and Number = 2. An equivalent wire diameter is displayed as Wire Size value in the Winding tab in the Properties window.

Stator Winding Data for Universal Motors

To access the stator winding data, double-click the **Machine-Stator-Winding** entry in the project tree.

The **Stator Winding Data Properties** window contains the following fields:

Insulation Thickness	The thickness of the insulation between the stator core and the field winding.
End Adjustment	The end length adjustment of the stator coils, which is the distance one end of the conductor extends vertically beyond the end of the stator.
Parallel Branches	The number of parallel branches in the stator winding.
Turns per Pole	The number of turns per stator pole (0 for auto-design).
Number of Strands	The number of wires per conductor (0 for auto-design).
Wire Wrap	The thickness of the double-sided wire wrap (0 to automatically obtain this value from the wire library).
Wire Size	The diameter of the wire (0 for auto-design). Click the button to open the **Wire Size** window where you can specify units, wire type, diameter, and gauge.

Defining the Rotor Data for a Universal Motor

The rotor is equipped with slots containing copper conductors that are connected to the commutator. The commutator acts as a mechanical rectifier in the motor.

The rotor consists of copper bars in which current is induced by the magnetic fields produced by the stator windings. In the project tree, double-click **Machine>Rotor** and **Machine-Rotor-Pole** to define the rotor and the pole.

To define general stator data:

1. To open the **Rotor Data Properties** window, double-click the **Machine>Rotor** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)

2. Enter the stacking factor for the rotor core in the **Stacking Factor** field. This value relates to the effective magnetic length of the core, and ranges from 0 to 1. It is defined as the total length minus the total insulation from the laminations, divided by the total length. A value of 1 indicates that the rotor is not laminated.
3. Enter the number of slots in the rotor core in the **Number of Slots** field.
4. Select a **Slot Type**:
   a. Click the button.
      The **Select Slot Type** window appears.
   b. Click a button to select the desired pole type (1, 2, 3, 4, 5, or 6). Though slots **3** and **4** are visually similar, they differ in how the edges are constructed. Slot **3** has a tapered edge leading from the slot opening to the main slot body. Slot **4** has a rounded edge at the same location, where the quantity **Hr1** defines the radius of the corner slot. TIP: When you run the mouse over each option, the diagram changes to show that pole type.
   c. Click **OK** to close the **Select Slot Type** window and return to the **Properties** window.

   Optionally, check **User Defined Slot** if you wish to define the slot dimensions using the **Slot Editor**.
5. Enter the outer diameter of the rotor core in the **Outer Diameter** field.
6. Enter the inner diameter of the rotor core in the **Inner Diameter** field.
7. Enter the length of the rotor core in the **Length** field.
8. Select a **Steel Type** for the rotor core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
9. Enter the number of slots in the skew width in the **Skew Width** field.
10. Click **OK** to close the **Properties** window.

Rotor Data for Universal Motors

To access the general rotor data, double-click the **Machine>Rotor** entry in the project tree.

The **Rotor Data Properties** window contains the following fields:

Stacking Factor	The effective magnetic length of the rotor core.
Number of Slots	The number of slots in the rotor core.
Slot Type	The rotor core slot type. Click the button to open the **Select Slot Type** window and select from the following types: **1**, **2**, **3**, **4**, **5**, **6**.
Outer Diameter	The outer diameter of the rotor core.

Inner Diameter	The inner diameter of the rotor core.
Length	The length of the rotor core.
Steel Type	The steel type of the rotor core. Click the button to open the **Select Definition** window.
Skew Width	The skew width measured in slot number.

Defining the Rotor Slots for Universal Motors

To define the physical dimensions of the rotor slots:

1. To open the **Rotor Slot Data Properties** window, double-click the **Machine-Rotor-Slot** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Optionally, to automatically design the dimensions of slots **Hs2**, **Bs1**, and **Bs2**, select the **Auto Design** check box. Using this option causes the software to converge to a flux density value of 1.5 Tesla in the rotor teeth.
3. Enter the available slot dimensions. The following dimensions may be listed, depending on the **Slot Type** selected and depending on whether or not **Auto Design** is selected.:

Hs0	Always available.
Hs1	Always available.
Hs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.
Bs0	Always available.
Bs1	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Bs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Rs	Always available. Rs is added when the slot type is 3 or 4.

4. Click **OK** to close the **Properties** window.

Rotor Slot Data for Universal Motors

To access the rotor slot data, double-click the **Machine-Rotor-Slot** entry in the project tree.

The **Rotor Slot Data Properties** window contains the following fields:

Auto Design	Select or clear this to enable or disable auto-design of slots **Hs2**, **Bs1**, and **Bs2**. When this check box is selected, only two other fields appear in the window: **Hs0** and **Bs0**.

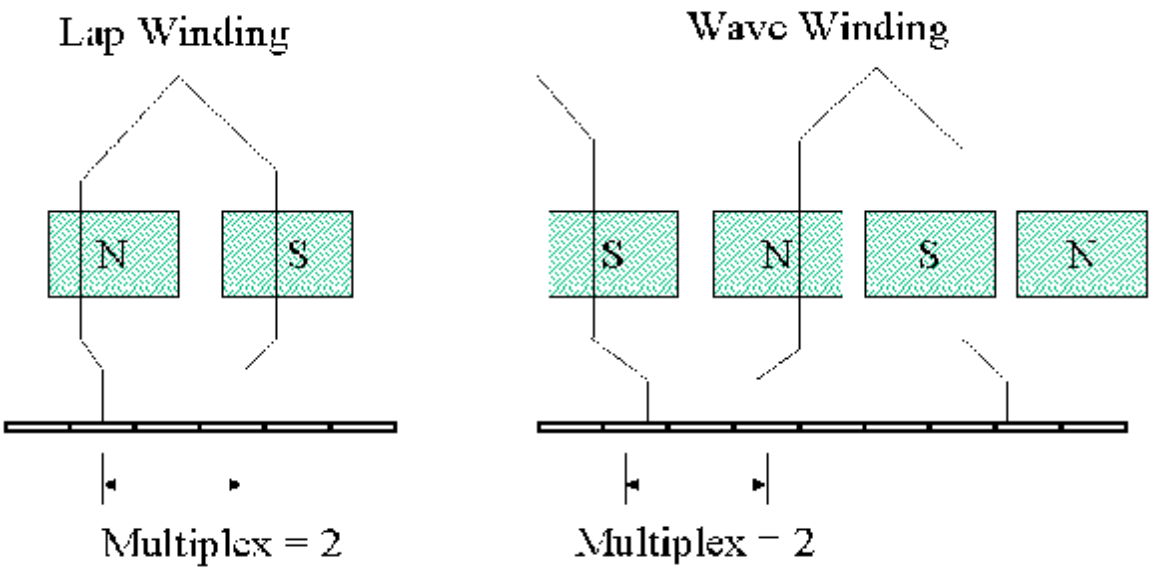
Parallel Tooth	Select this to design **Bs1** and **Bs2** based on the tooth width. When this check box is selected, the **Bs1** and **Bs2** fields are removed, and the **Tooth Width** field is added.
Tooth Width	The tooth width for the parallel tooth, on which **Bs1** and **Bs2** are designed.
Hs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Hs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Hs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Bs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Bs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Bs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Rs	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Rs	A slot dimension. (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected). Rs is added when the slot type is 3 or 4.

Defining the Rotor Windings and Conductors for a Universal Motor

To define the rotor windings, wires, and conductors:

1. To open the **Rotor Slot Winding Properties** window, double-click the **Machine-Rotor-Winding** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Click the **Winding** tab.
3. Select a **Winding Type**:
   a. Click the button for **Winding Type**.
   The **Winding Type** window appears.
   b. Select from one of the following three types of winding:
      - **Lap**
      - **Wave**
      - **Frog Leg**
4. Enter the number of windings in the **Multiplex Number** field (1 for a single winding, 2 for double windings, 3 for triple windings). For a lap winding, the multiplex number is the number of commutators between the start and end of one winding, and the number of parallel branches is equal to the number of poles multiplied by the multiplex number. For a wave winding, the number of parallel branches equals the multiplex number multiplied by

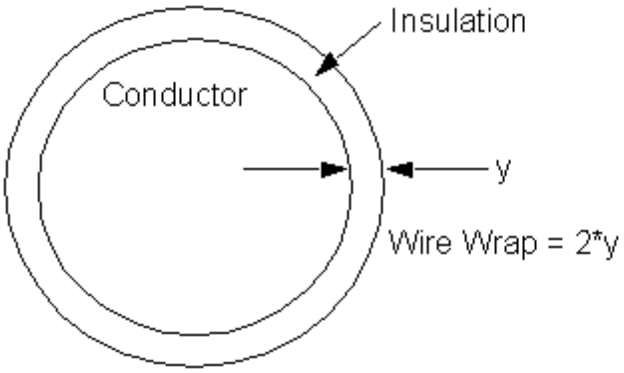
two.



5. Enter the number of virtual slots per each real slot in the **Virtual Slots** field. The rotor is assumed to have two layers of conductors, an upper and a lower layer. Each layer of conductors can have a number of windings, which are referred to as virtual slots.

Note	For example, the upper and lower layer can have two windings each, which would yield a virtual slot number of two; for a 12 slot machine, this would yield 24 commutation segments.

6. Enter the total number of conductors in each rotor slot in the **Conductors per Slot** field. This value is the number of turns per coil multiplied by the number of layers. This value is the total number of conductors in one real full rotor slot.
7. Enter the coil pitch, measured in number of slots, in the **Coil Pitch** field. The coil pitch is the number of slots separating one winding. For example, if a coil starts in slot 1 and ends in slot 6, it has a coil pitch of 5.
8. Enter the number of wires per conductor in the **Number of Strands** field. Enter 0 to have RMxprt auto-design this value.
9. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



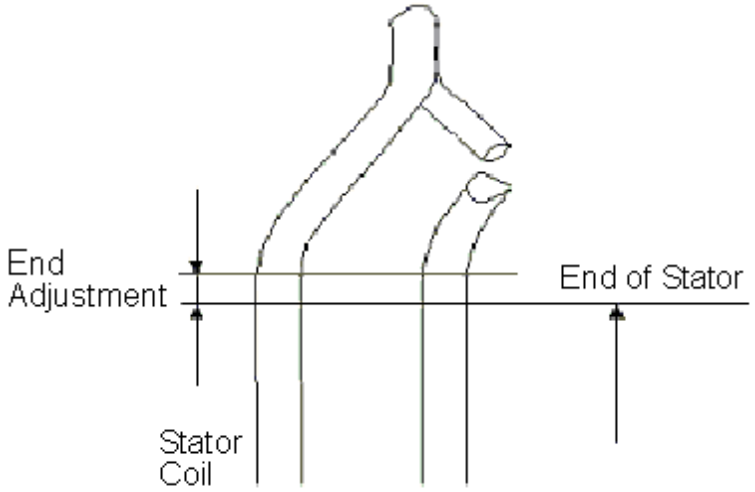
10. Select the **Wire Size**:
    a. Click the button for **Wire Size**.
    The **Wire Size** window appears.
    b. Select a value from the **Wire Diameter** pull-down list.
    c. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

< ***number*** **>**	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.
USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.
AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.
MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21mm and 2 with a diameter of 0.13mm.

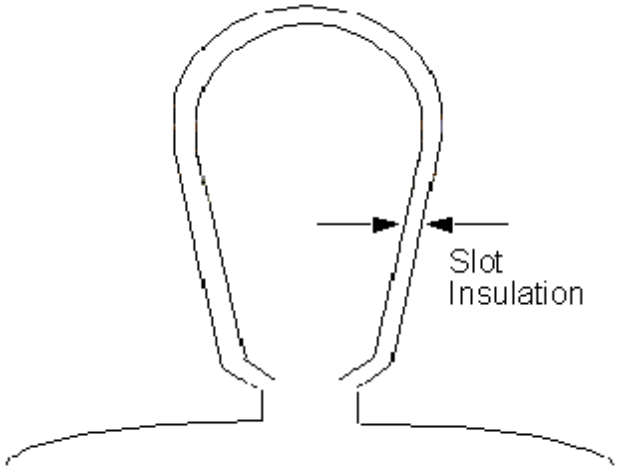
The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>General Options>Machines**command. When you are done setting the wire size, click **OK** to close the **Wire Size** window and return to the **Properties** window. Click the **End/Insulation** tab. Select or clear the **Input Half-turn Length** check box. Do one of the following:

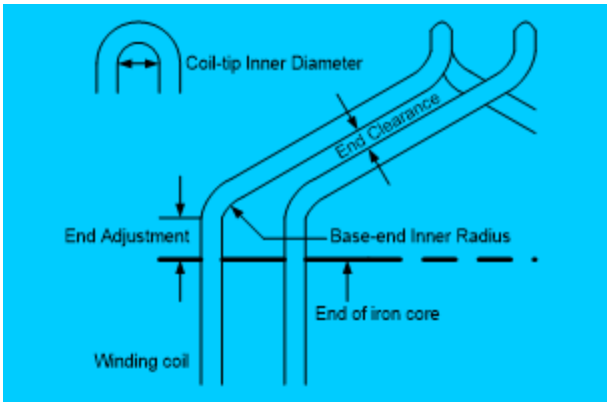
- If you selected **Input Half-turn Length**, then enter the half-turn length of the armature winding in the Half Turn Length field.
- If you cleared **Input Half-turn Length**, then enter the end length adjustment of the stator coils in the **End Adjustment** field. The end adjustment is the distance one end of the

conductor extends vertically beyond the end of the stator.



11. Enter the inner radius of the base corner in the **Base Inner Radius** field.
12. Enter the inner diameter of the coil tip in the **Tip Inner Diameter** field.
13. Enter the distance between two rotor coils in the **End Clearance** field.
14. Enter the thickness of the slot liner insulation in the **Slot Liner** field.





15. Enter the thickness of the wedge insulation in the **Wedge Thickness** field.
16. Enter the thickness of the insulation layer in the **Layer Insulation** field.
17. Enter the limited slot fill factor for the wire design in the **Limited Fill Factor** field. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.
18. Select the type of equalizer connection from the **Equalizer Connection** pull-down menu. Select from **None**, **Half**, or **Full**.
19. Click **OK** to close the **Properties** window.

Defining Different Size Wires for a Universal Motor Rotor Winding

Use the **Gauge** option if you have a conductor that is made up different size wires.

To define different size wires:

1. In the **Wire Size** window, select **MIXED** from the **Gauge** pull-down menu.
2. Select either **Round** or **Rectangular** as the **Wire Type**.
3. Enter the appropriate wire data in the table:
   - For a round wire:
     - Enter the **Diameter** in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this diameter.
   - For a rectangular wire:
     - Enter the **Width** of the wire in the table.
     - Enter the **Thickness** of the wire in the table.
     - Enter the **Fillet** value in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this data.
4. Click **Add** to add the new wire data.
5. Repeat steps 3 and 4 for each size wire you want to add.

6. When you are finished defining the wires, click **OK** to close the **Wire Size** window and return to the RMxprt **Properties** window.

Note	For example, if one conductor is made up of 5 wires, and 3 of those wires have a diameter of 0.21mm, and the other 2 have a diameter of 0.13mm, then the mixed wire size table will have two lines. The first line will list Diameter = 0.21 and Number = 3. The second line will list Diameter = 0.13 and Number = 2. An equivalent wire diameter is displayed as Wire Size value in the Winding tab in the Properties window.

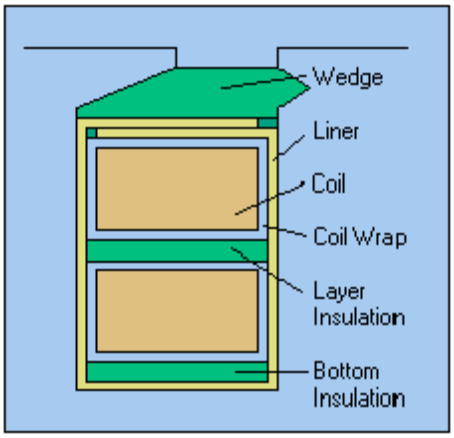
Rotor Winding Data for Universal Motors

To access the stator winding data, double-click the **Machine-Rotor-Winding** entry in the project tree.

The **Rotor Winding Data Properties** window contains the following fields:

Winding tab	**Winding Type**	The type of rotor winding. Click the button to open the **Winding Type** window and choose from **Lap**, **Wave**, and **Frog Leg**.
	Multiplex Number	Single, double, or triple windings (**1**, **2**, or **3**).
	Virtual Slots	The number of virtual slots per real slot.
	Conductors per Slot	The number of conductors per rotor slot (0 for auto-design).
	Coil Pitch	The coil pitch measured in number of slots.
	Number of Strands	The number of wires per conductor (0 for auto-design).
	Wire Wrap	The thickness of the double-sided wire wrap (0 to automatically obtain this value from the wire library).
	Wire Size	The diameter of the wire (0 for auto-design). Click the button to open the **Wire Size** window where you can specify units, wire type, diameter, and gauge.
End/Insulation tab	**Input Half-turn Length**	Select or clear this check box to specify whether or not you want to enter the half-turn length. When this check box is selected, the **Half Turn Length** field appears the next time you

		open the **Properties** window. When this check box is selected, the **End Adjustment** field appears instead.
	Half Turn Length	The half-turn length of the armature winding.
	End Adjustment	The end length adjustment of the rotor coils.
	Base Inner Radius	The inner radius of the base corner.
	Tip Inner Diameter	The inner diameter of the coil tip.
	End Clearance	The end clearance between two adjacent coils.
	Slot Liner	The thickness of the slot liner insulation.
	Wedge Thickness	The thickness of the wedge insulation.
	Layer Insulation	The thickness of the insulation layer.
	Limited Fill Factor	The limited slot fill factor for the wire design. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.
	Equalizer Connection	The connection type of the equalizer. Select from **None**, **Half**, or **Full**.



Defining the Commutator and Brush for a Universal Motor

The commutator allows current transfer between DC terminals or brushes and the rotor coils, providing the current to the system as a function of rotation. Due to the action of the commutator,

the corresponding magnetic field has a fixed distribution with respect to the stator.

To define the commutator and brush pairs:

1. To open the **Commutator Data Properties** window, double-click the **Machine>Commutator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Click the **Commutator** tab.
3. Select **Cylinder** or **Pancake Type** as the **Commutator Type**.

Note	When you place the mouse cursor over the commutator type, an outline of the commutator appears.

4. For **Cylinder** commutators, do the following:
   a. Enter the **Commutator Diameter**.
   b. Enter the **Commutator Length**.
5. For **Pancake** commutators, do the following:
   a. Enter the **Outer Diameter**.
   b. Enter the **Inner Diameter**.
6. Enter the thickness of the insulation between two consecutive commutator segments in the **Commutator Insulation** field.
7. Click the **Brush** tab.
8. Enter the **Brush Width**.
9. Enter the **Brush Length**.
10. Enter the number of brush pairs when using a wave armature winding in the **Brush Pairs** field.
11. Enter the angle of displacement from the neutral axis, in mechanical degrees, in the **Brush Displacement** field.

Note	The brush displacement is positive for the counter-clockwise direction. For example, if the rotor turns clockwise and the brush displacement is also clockwise, then the angle is negative; if the rotor turns clockwise but the brush displacement is counter-clockwise, then the angle is positive.

12. Enter the voltage drop across one brush pair in the Brush Drop field.
13. Enter the mechanical pressure of the brushes as they press against the commutator in the **Brush Press** field.
14. Enter the **Frictional Coefficient** of the brush.

Note	If the **Friction Loss** field is used in the **General** window, the **Brush Press** and **Frictional Coefficient** fields will be hidden in the **Commutator/Brush** window. These fields are shown only when the **Friction Loss** field in the **General** window is set to zero.

15. Click **OK** to close the **Properties** window.

Commutator and Brush Data for Universal Motors

To access the commutator and brush data, double-click the **Machine>Commutator** entry in the project tree.

The **Commutator Data Properties** window contains the following fields:

Commutator tab	**Commutator Type**	The type of commutator. Click the button to open the **Select Commutator Type** window and select from **Cylinder** or **Pancake**.
	Commutator Diameter	For a **Cylinder** commutator type, the diameter of the commutator.
	Commutator Length	For a **Cylinder** commutator type, the length of the commutator.
	Outer Diameter	For a **Pancake** commutator type, the outer diameter of the commutator.
	Inner Diameter	For a **Pancake** commutator type, the inner diameter of the commutator.
	Commutator Insulation	The thickness of the insulation between the two commutator bars.
Brush tab	**Brush Width**	The width of the brush.
	Brush Length	The length of the brush.
	Brush Pairs	The number of brush pairs.
	Brush Displacement	The displacement of the brush from the neutral position, in mechanical degrees (positive for anti-rotating direction).
	Brush Drop	The voltage drop across a one-pair brush.
	Brush Press	The brush press per unit area. (Available only when **Frictional Loss** is set to zero for the machine.)
	Frictional Coefficient	The frictional coefficient of the brush. (Available only when **Frictional Loss** is set to zero for the machine.)

Defining the Shaft Data for a Universal Motor

To define the shaft:

1. To open the **Shaft Data Properties** window, double-click the **Machine>Shaft** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select or clear the **Magnetic Shaft** check box to specify whether or not the shaft is to be made of magnetic material.
3. Click **OK** to close the **Properties** window.

Shaft Data for Universal Motors

To access the shaft data, double-click the **Machine>Shaft** entry in the project tree.

The **Shaft Data Properties** window contains the following fields:

Magnetic Shaft	Select or clear this check box to indicate whether or not the shaft is made of magnetic material.

Setting Up Analysis Parameters for a Universal Motor

To define the solution data:

1. To open the **Solution Setup** window, right-click **Analysis** in the project tree, and click **Add Solution Setup**.
2. Click the **General** tab. The **Operation Type** is automatically set to **Motor** for this machine type.
3. Select the **Load Type** used in the motor from the following options:

Const Speed	The speed remains constant in the motor.
Const Power	The output power remains constant in the motor.
Const Torque	The torque remains constant regardless of the speed. In this case, $T_{load} = T_{rated}$, given by the output power divided by the given rated speed.
Linear Torque	The torque increases linearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})$ where T_{rated} is given by the output power divided by the given rated speed.
Fan Load	The load varies nonlinearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})^2$ where T_{rated} is given by the output power divided by the given rated speed.

4. Enter the output power developed at the shaft of the motor in the **Rated Output Power** field.
5. Enter the RMS line-to-line voltage in the **Rated Voltage** field.
6. Enter the desired output speed of the motor at the load point in the **Rated Speed** field.
7. Enter the temperature at which the system functions in the **Operating Temperature** field.
8. Click the **Universal Motor** tab.
9. Enter the **Frequency**, and select the units.
10. Click **OK** to close the **Solution Setup** window.

Related Topics

Solution Data for Universal Motors

Solution Data for Universal Motors

To access the solution data, right-click **Analysis** in the project tree, and click **Add Solution Setup**. For this machine type, there is only one tab, the **General** tab.

The **Solution Setup** window contains the following fields:

Operation Type	***General** tab*. The operation type is automatically set to **Motor** for this machine type.
Load Type	***General** tab*. Select from **Const Speed**, **Const Power**, **Const Torque**, **Linear Torque**, and **Fan Load**. The default is **Const Power**.
Rated Output Power	***General** tab*. Type a value for the rated output voltage, and select the units.
Rated Voltage	***General** tab*. Type a value for the rated voltage, and select the units.
Rated Speed	***General** tab*. Type a value for the rated speed, and select the units.
Operating Temperature	***General** tab*. Type a value for the operating temperature, and select the units.
Frequency	***Universal Motor** tab.* Enter a frequency in the **Frequency** field, and select the units.

Related Topics

Setting Up Analysis Parameters for a Universal Motor

General DC Machines

After you have selected **DC Machine** as your model type, enter the motor data to define the following:

- General data, such as the output power, rated voltage, speed, and machine type (motor or generator).
- Stator data, such as its associated pole dimensions, type of steel, and pole magnet specifications.
- Stator field data, such as shoe and pole insulation, dimensions, and winding information.
- Rotor data, such as the slot types and dimensions, rotor diameter, lamination, and wire specifications.
- Commutator and brush data, such as the commutator type and dimensions and brush length.
- Shaft data.
- Solution data.

By option you can insert or remove the following for a DC machine.

- Add a machine housing.
- Compensating data, added under the stator

- Commutating data, added under the stator
- Vent data, added under the rotor
- Shunt data, added under the stator field.
- Series data, added under the stator field.

Analysis Approach for General DC Machines

For a Direct-Current (DC) Electric Machine Design, either a generator or motor, the rotor is equipped with a distributed winding -- called armature winding -- that is connected to a commutator revolving together with the rotor.

The stator is equipped with *p* pairs of poles, which are excited by *p* pairs of shunt and/or series windings. A shunt winding may be separately excited or self-excited. The separately excited shunt winding is excited by a separate DC voltage source. The self-excited shunt winding is excited by the terminal voltage of the armature winding and is connected in parallel with the armature winding. A series winding is connected in series with the armature winding. If both self-excited shunt and series windings are mounted on the stator poles, RMxprt assumes that the armature winding connects the series winding in series first, then connects the shunt winding in parallel.

A system of brushes is kept in permanent electrical contact with the commutator. When DC current is applied to the rotating armature winding via the brushes and commutator, a stationary magnetic field distribution is created with the axis electrically perpendicular to the axis of the field produced by the shunt and/or series windings. As a result, a torque is produced by the interaction of the fields produced by the armature and exciting currents. For these brush commutating machines, the commutator together with the brushes acts as a mechanical rectifier.

The field produced by the armature current is called armature reaction field. The armature reaction field causes poor commutating and poor voltage distribution along commutator bars. In order to improve commutating, commutating poles and winding can be equipped between two adjacent main poles and compensating winding can be equipped under main poles.

The performance of a DC machine is computed by DC analysis.

DC Machine Operating as a Motor

The voltage equation of a DC motor is

$$U = E + (U_b + R_1 \cdot I_a)$$

where, U_b is the voltage drop of one-pair brushes, R_1 is the total series resistance of the armature branch, E is the back emf as given below:

$$E = C_{Ef} \cdot \omega \cdot I_f + C_{Es} \cdot \omega \cdot I_a$$

where C_{Ef} and C_{Es}, which depend on the saturation of the magnetic field, are the back-emf coefficients in ohm.s/rad, is the rotor speed in mechanical rad/s, and I_f and I_a are the exciting currents of the shunt and series windings, respectively.

For a given speed, armature current can be computed based on the terminal voltage U, as shown below:

$$I_a = \frac{U - U_b - C_{Ef} \cdot \omega \cdot I_f}{R_1 + C_{Es} \cdot \omega}$$

The shaft torque is computed from:

$$T_2 = (C_{Tf} \cdot I_f + C_{Ts} \cdot I_a) \cdot I_a - T_{fw}$$

where C_{Tf} and C_{Ts} are the torque coefficients in Nm/A^2 which are numerically the same as C_{Ef} and C_{Es}, respectively. T_{fw} is the frictional and wind torque. The output power (mechanical power) is

$$P_2 = T_2 \cdot \omega$$

The input power (electrical power) is

$$P_1 = P_2 + (P_{fw} + P_{Cua} + P_b + P_{Fe})$$

where P_{fw}, P_{Cua}, P_b, and P_{Fe} are the frictional and wind loss, armature branch copper loss, brush drop loss, iron-core loss and shunt winding copper loss, respectively. The efficiency is:

$$\eta = \frac{P_2}{P_1} \times 100\ \%$$

DC Machine Operating as a Generator

For a DC generator, the voltage equation is

$$U = E - (U_b + R_1 \cdot I_a)$$

$$E = C_{Ef} \cdot \omega \cdot I_f + C_{Es} \cdot \omega \cdot I_a$$

The performance is analyzed as follows

$$I_a = -\frac{U + U_b - C_{Ef} \cdot \omega \cdot I_f}{R_1 - C_{Es} \cdot \omega}$$

$$T_1 = ((C_{Tf} \cdot I_f + C_{Ts} \cdot I_a) \cdot I_a) + T_{fw}$$

$$P_1 = T_1 \cdot \omega$$

$$P_2 = P_1 - (P_{fw} + P_{Cua} + P_b + P_{Fe})$$

$$\eta = \frac{P_2}{P_1} \times 100\,\%$$

Defining a General DC Machine

The general procedure for defining a general DC machine is as follows:

1. Insert a DC machine into a new or existing project.
2. Double-click the **Machine** entry in the project tree to define the general data.
3. Double-click the **Machine-Stator** entry in the project tree to define the stator geometry.
4. Double-click the **Machine-Stator-Pole** entry in the project tree to define the stator pole dimensions.
5. Double-click the **Machine-Stator-Field** entry in the project tree to define the stator windings, conductors, and insulation data.
6. Double-click the **Machine-Rotor** entry in the project tree to define the general rotor geometry, the pole data, and the insulation data.
7. Double-click the **Machine-Rotor-Slot** entry in the project tree to define the rotor slot dimensions.
8. Double-click the **Machine-Rotor-Winding** entry in the project tree to define the rotor conductors and windings.
9. Double-click the **Machine-Commutator** entry in the project tree to define the commutator and brush data.
10. Double-click the **Machine-Shaft** entry in the project tree to define the magnetism of the shaft.
11. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
12. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.

13. Choose **File>Save** to save the project.
14. Choose **RMxprt>Analyze** to analyze the design.

Note	When you place the cursor over an entry field, a brief description of that field appears in the status bar at the bottom of the **RMxprt** window.

Once analyzed, the model can be viewed in the Maxwell 2D Modeler, or it can be used to create a new Maxwell 2D project, and a new Maxwell 3D design.

Refer to the *DC Machine* application note, on the technical support page of the Ansys web site, for a specific example of a problem using a DC machine.

Defining the General Data for a General DC Machine

Use the **General** window to define the basic parameters of the DC motor, such as the power settings, speed, and rated voltage.

To define the general data:

1. To open the **General Data Properties** window, double-click the **Machine** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the number of poles for the machine in the **Number of Poles** field. This value is the total number of poles in the stator (or the number of pole pairs multiplied by two).
3. Enter the energy loss due to friction at the given speed in the **Frictional Loss** field.
4. Enter the wind loss due to air resistance measured at the reference speed in the **Wind Loss** field.
5. Enter the given speed in the **Reference Speed** field.
6. Click **OK** to close the **Properties** window.

General Data for General DC Machines

To access the general data, double-click the **Machine** entry in the project tree.

The **General Data Properties** window for a three-phase induction motor contains the following fields:

Machine Type	The machine type you selected when inserting a new RMxprt design (DC Machine).
Number of Poles	The number of poles the machine contains.
Frictional Loss	The frictional energy loss (due to friction) measured at the reference speed.
Wind Loss	The wind loss (due to air resistance) measured at the reference speed.
Reference Speed	The given speed of reference.

Defining the Stator Data for a General DC Machine

Use the **Stator Properties** windows to define the stator dimensions, slots, windings, and conductors.

The stator is the outer lamination stack where the polyphase voltage windings reside.

To define the general stator data:

1. To open the **Stator Data Properties** window, double-click the **Machine>Stator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the maximum diameter for a polygon-type frame in the **Frame Outer Diameter** field.
3. Enter the minimum outer width for a polygon-type frame in the **Frame Overall Width** field.
4. Enter the **Frame Thickness**.
5. Enter the **Frame Length**.
6. Select a steel type for the frame:
   a. Click the button for **Frame Material**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
7. Select a **Pole Type**:
   a. Click the button.
      The **Select Pole Type** window appears.
   b. Click a button to specify the desired field type (either 1 or 2).
   c. Click **OK** to close the **Select Pole Type** window and return to the Properties window.
8. Enter the length of the stator main pole in the **Pole Length** field.
9. Enter the effective magnetic length for the stator main pole in the **Pole Stacking Factor** field.
10. Select a steel type for the stator main pole:
    a. Click the button for **Pole Material**.
       The **Select Definition** window appears.
    b. Select a steel type from the list, or define a new steel type.
    c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
11. Enter the thickness of the pole press boards in the **Press Board Thickness** field.
12. If the pole press board is made of magnetic material, then select the **Magnetic Press Board** check box.
13. Click **OK** to close the **Properties** window.

Stator Data for General DC Machines

To access the general stator data, double-click the **Machine>Stator** entry in the project tree.

The **Stator Data Properties** window contains the following fields:

Frame Outer Diameter	The maximum diameter for a polygon-type frame.
Frame Overall Width	The minimum outer width for a polygon-type frame.
Frame Thickness	The thickness of the frame.
Frame Length	The length of the frame.
Frame Material	The steel type of the frame. Click the button to open the **Select Definition** window.
Pole Type	The pole type of the stator. Click the button to open the **Select Pole Type** window and select from the following two types: **1** and **2**.
Pole Length	The length of the stator main pole.
Pole Stacking Factor	The stacking factor of the stator main pole.
Pole Material	The steel type of the stator main pole. Click the button to open the **Select Definition** window.
Press Board Thickness	The thickness of the pole press boards.
Magnetic Press Board	Whether or not the pole press board is made of magnetic material.

Defining the Stator Pole for a General DC Machine

The rotor pole drives the electromagnetic field which is coupled with the stator windings. Use the **s Stator Pole Data Properties** window to define the stator pole.

To define the rotor pole:

1. To open the **Stator Pole Data Properties** window, double-click the **Machine-Stator-Pole** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)

Note	For a two-pole machine, a pole embrace of 0.75 yields a magnet with a span of 135 degrees (based on 0.75*180 degrees).

2. Enter the inner diameter at the pole center in the Dmin field.
3. Enter the diameter at the pole tip in the Dmax field.
4. Enter the width of the pole arc with a uniform air gap in the Bp0 field. For an eccentric air gap, enter 0.
5. Enter the width of the pole tip in the Bp1 field.

6. Enter the maximum width of the pole shoe in the Bp2 field. This field is only available for a Pole Type of 1.
7. Enter the minimum width of the pole shoe in the Bp3 field. This field is only available for a Pole Type of 1.
8. Enter the size of the pole shoe fillet in the Rp0 field. THis field is only available for a Pole Type of 2.
9. Enter the fillet between the pole shoe and the pole body in the Rp1 field. THis field is only available for a Pole Type of 2.
10. Enter the pole shoe height in the Hp field.
11. Enter the pole body width in the Bm field.
12. Click **OK** to close the **Properties** window.

Stator Pole Data for General DC Machines

To access the stator pole data, double-click the **Machine-Stator-Pole** entry in the project tree.

The **Rotor Pole Data Properties** window contains the following fields:

Dmin	The inner diameter at the pole center.
Dmax	The diameter at the pole tip.
Bp0	The width of the pole arc with a uniform air gap (0 for an eccentric air gap).
Bp1	The width of the pole tip.
Bp2	The maximum width of the pole shoe. For pole type 1.
Bp3	The minimum width of the pole shoe. FOr pole type 1.
Rp0	The pole shoe fillet. For pole type 2.
Rp1	The fillet between the pole shoe and the pole body. For pole type 2.
Hp	The height of the pole shoe.
Bm	The width of the pole body.

Defining the Stator Field Data for a General DC Machine

To define the stator windings and insulation data:

1. To open the **Stator Field Properties** window, double-click the **Machine-Stator-Field** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the thickness of the insulation under the pole shoe in the **Shoe Insulation** field.
3. Enter the thickness of the insulation at the pole body side in the **Pole Insulation** field.
4. Enter the minimum gap in the **Winding Clearance** field. The winding clearance is one of the following: the minimum gap between two field windings, or the minimum gap between a field winding and a commutating winding.

5. Enter the thickness of the insulation between the shunt winding and the series winding in the **Winding Insulation** field.
6. Select the type of exciting of the series winding to the shunt winding from the **Compound Exciting Mode** pull-down list. The options are **Cumulative** and **Differential**.
7. Click **OK** to close the **Properties** window.

Stator Field Data for General DC Machines

To access the stator field data, double-click the **Machine-Stator-Field** entry in the project tree.

The **Stator Field Data Properties** window contains the following fields:

Shoe Insulation	The thickness of the insulation under the pole shoe.
Pole Insulation	The thickness of the insulation at the pole body side.
Winding Clearance	The minimum air gap between two field windings, or the minimum gap between a field winding and a commutating winding.
Winding Insulation	The thickness of the insulation between the shunt winding and the series winding.
Compound Exciting Mode	The cumulative exciting or differential exciting of the series winding to the shunt winding. Select **Cumulative** or **Differential** from the pull-down list.

Shunt Data for General DC Machines

By option you can insert or remove a shunt from a General DC Machine. If you insert a shunt, it appears in the project tree under the stator field data.

To insert a shunt.

1. Right click on the Field icon under the stator in the project tree to display the popup menu.
2. Click **Insert Shunt**.

   The Shunt icon appears under the field icon.

To Remove an existing shunt:

1. Right click on the Field icon under the stator in the project tree to display the popup menu.
2. Click **Remove Shunt**.

   The shut is removed from the project tree.

The Shunt data for a General DC Machine contains the following fields.

Winding type	Specified as Round, Cylinder coil, or Edgewise coil, by clicking the button to display the Winding Type selection window.
Parallel	Number of parallel branches.

branches	
Conductors per pole	Number of conductors per pole. 0 for auto-design. Odd number of strands for the case where the input and output leads are on different sides.
Number of strands	Number of strands (number of wires per conductor). 0 for auto-design.
Wire wrap	Double side wire wrap thickness. 0 for auto-pickup in the wire library.
Wire size.	Click the button to display the Wire Size selection window.
Axial Clearance	Axial gap between field winding and pole body on the inner coil.
Limited cross width	Limited cross-section width for winding design or arrangement. 0 for available maximum area.
Limited cross height	Limited cross-section height for winding design or arrangement. 0 for available maximum area.
Winding fillet.	

Series Data for General DC Machines

By option, you can insert or remove a series from a General DC Machine. If you insert a series, it appears in the project tree under the stator field data.

To insert a series:

1. Right click on the Field icon under the stator in the project tree to display the popup menu.
2. Click **Insert Series**.

   The Series icon appears under the field icon.

To Remove an existing series:

1. Right click on the Field icon under the stator in the project tree to display the popup menu.
2. Click **Remove Series**.

   The series is removed from the project tree.

The Series data for a General DC Machine contains the following fields.

Winding type	Specified as Round, Cylinder coil, or Edgewise coil, by clicking the button to display the Winding Type selection window.
Parallel branches	Number of parallel branches.
Conductors per pole	Number of conductors per pole. 0 for auto-design. Odd number of strands for the case where the input and output leads are on different sides.
Number of	Number of strands (number of wires per conductor). 0 for auto-design.

strands	
Wire wrap	Double side wire wrap thickness. 0 for auto-pickup in the wire library.
Wire size.	Click the button to display the Wire Size selection window.
Axial Clearance	Axial gap between field winding and pole body on the inner coil.
Limited cross width	Limited cross-section width for winding design or arrangement. 0 for available maximum area.
Limited cross height	Limited cross-section height for winding design or arrangement. 0 for available maximum area.
Winding fillet.	

Compensating Data for General DC Machines

By option, you can insert or remove Compensating for a General DC Machine.

To insert compensating:

1. Right-click on the Stator icon to display the pop-up menu.
2. Click **Insert Compensating**.

To remove an existing Compensating:

1. Right click on the Stator icon to display the pop-up menu.
2. Click **Remove Compensating**.

To access the data for compensating inserted to a General DC Machine, double click on the **Machine-Rotor-Compensating** item in the project tree.

The **Compensating** properties window contains the following fields.

Slots per pole	Number of slots per pole for the compensating winding.
Bc0	Opening width of the compensating slots.
Hc0	Opening height of the compensating slots.
Bc2	Width of the compensating slots.
Hc2	Height of the compensating slots.
Parallel branches	Number of parallel branches.
Conductors per slot	Number of conductors per slot for the compensating windings
Number of strands	Number of strands (number of wires per conductor), 0 for auto-design.

Wire wrap	Double-side wire wrap thickness, 0 for auto pickup in the wire library
Rectangle wire	Whether to use round (the default) or rectangle wire.
Wire size	Click the button to display the Wire Size window to specify the wire diameter and gauge.
Slot liner	Insulation slot liner thickness
End adjustment	one side end length adjustment of a conductor.

Commutating Data for General DC Machines

Commutating must be inserted under the stator by right-clicking on the stator icon to display the pop-up menu, and click **Insert Commutating** command. This command also inserts an icon in the project tree for an associated winding.

To remove an existing Commutating (and associated winding), right-click on the stator icon to display the pop-up menu and click **Remove Commutating**. This removes the commutating and the associated winding.

Note: This is distinct from the general Commutator data associated with rotor.

Pole width	Width of the commutating poles
Pole height	Height of the commutating poles.
Pole length	Length of the commutating poles
Shoe width	Shoe width of the commutating poles
Shoe height	Shoe height of the commutating poles.
Second air gap	Length of the second air gap between the commutating pole and the frame.
Pole stacking factor	Stacking factor for the commutating poles.
Pole material	Steel type of the commutating poles. Click the button to display the Select Definition window.
Pole insulation	Thickness of insulation on the pole body side.

Winding Data for Commutating

If you have inserted commutating for a General DC machine, an additional winding icon appears in the project tree for the associated winding.

Winding type	Specified as Round, Cylinder coil, or Edgewise coil, by clicking the button to display the Winding Type selection window.
Parallel branches	Number of parallel branches.

Conductors per pole	Number of conductors per pole. 0 for auto-design. Odd number of strands for the case where the input and output leads are on different sides.
Number of strands	Number of strands (number of wires per conductor). 0 for auto-design.
Wire wrap	Double side wire wrap thickness. 0 for auto-pickup in the wire library.
Wire size.	Click the button to display the Wire Size selection window.
Axial Clearance	Axial gap between field winding and pole body on the inner coil.
Limited cross width	Limited cross-section width for winding design or arrangement. 0 for available maximum area.
Limited cross height	Limited cross-section height for winding design or arrangement. 0 for available maximum area.
Winding fillet.	

Defining the Rotor Data for a General DC Machine

The rotor is equipped with slots containing copper conductors that are connected to the commutator. The commutator acts as a mechanical rectifier in the motor.

The rotor consists of copper bars in which current is induced by the magnetic fields produced by the stator windings. In the project tree, double-click **Machine>Rotor** and **Machine-Rotor-Pole** to define the rotor and the pole.

To define general stator data:

1. To open the **Rotor Data Properties** window, double-click the **Machine>Rotor** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the stacking factor for the rotor core in the **Stacking Factor** field. This value relates to the effective magnetic length of the core, and ranges from 0 to 1. It is defined as the total length minus the total insulation from the laminations, divided by the total length. A value of 1 indicates that the rotor is not laminated.
3. Enter the number of slots in the rotor core in the **Number of Slots** field.
4. Select a **Slot Type**:
   a. Click the button.
      The **Select Slot Type** window appears.
   b. Click a button to select the desired slot type (1, 2, 3, 4, 5, or 6). Though slots **3** and **4** are visually similar, they differ in how the edges are constructed. Slot **3** has a tapered edge leading from the slot opening to the main slot body. Slot **4** has a rounded edge at the same location, where the quantity **Hr1** defines the radius of the corner slot. TIP: When

you run the mouse over each option, the diagram changes to show that pole type.

c. Click **OK** to close the **Select Slot Type** window and return to the **Properties** window.

Optionally, check **User Defined Slot** if you wish to define the slot dimensions using the **Slot Editor**.

5. Enter the number of lamination sectors in the **Lamination Sectors** field.
6. Enter the outer diameter of the rotor core in the **Outer Diameter** field.
7. Enter the inner diameter of the rotor core in the **Inner Diameter** field.
8. Enter the length of the rotor core in the **Length** field.
9. Select a **Steel Type** for the rotor core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
10. Enter the thickness of the pole press boards in the **Press Board Thickness** field.
11. Enter the number of slots in the skew width in the **Skew Width** field.
12. Click **OK** to close the **Properties** window.

Rotor Data for General DC Machines

To access the general rotor data, double-click the **Machine>Rotor** entry in the project tree.

The **Rotor Data Properties** window contains the following fields:

Stacking Factor	The effective magnetic length of the rotor core.
Number of Slots	The number of slots the rotor core contains.
Slot Type	The type of slots in the rotor core. Click the button to open the **Select Slot Type** window.
Lamination Sectors	The number of lamination sectors.
Outer Diameter	The outer diameter of the rotor core.
Inner Diameter	The inner diameter of the rotor core.
Length	The length of the rotor core.
Steel Type	The steel type of the rotor core. Click the button to open the **Select Definition** window.
Press Board Thickness	The thickness of the pole press boards.
Skew Width	The skew width measured in slot number.

Defining the Rotor Slots for a General DC Machine

To define the physical dimensions of the rotor slots:

1. To open the **Rotor Slot Data Properties** window, double-click the **Machine-Rotor-Slot** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Optionally, to automatically design the dimensions of slots **Hs2**, **Bs1**, and **Bs2**, select the **Auto Design** check box. Using this option causes the software to converge to a flux density value of 1.5 Tesla in the rotor teeth.
3. Enter the available slot dimensions. The following dimensions may be listed, depending on the **Slot Type** selected and depending on whether or not **Auto Design** is selected.:

Hs0	Always available.
Hs1	Always available.
Hs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.
Bs0	Always available.
Bs1	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Bs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Rs	Always available.

4. Click **OK** to close the **Properties** window.

Rotor Slot Data for General DC Machines

To access the rotor slot data, double-click the **Machine-Rotor-Slot** entry in the project tree.

The **Rotor Slot Data Properties** window contains the following fields:

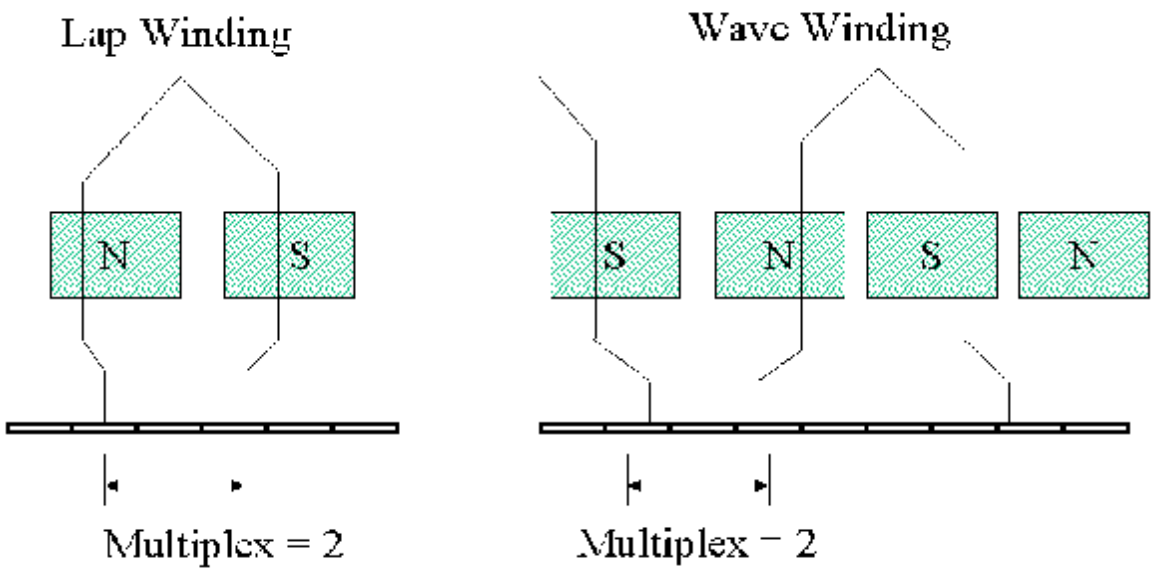
Auto Design	Select or clear this to enable or disable auto-design of slots **Hs2**, **Bs1**, and **Bs2**. When this check box is selected, only two other fields appear in the window: **Hs0** and **Bs0**.
Parallel Tooth	Select this to design **Bs1** and **Bs2** based on the tooth width. When this check box is selected, the **Bs1** and **Bs2** fields are removed, and the **Tooth Width** field is added.
Tooth Width	The tooth width for the parallel tooth, on which **Bs1** and **Bs2** are designed.
Hs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Hs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Hs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Bs0	A slot dimension (see the diagram shown in the modeling window when **Machine-**

	Rotor-Slot is selected).
Bs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Bs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Rs	A slot dimension (see the diagram shown in the modeling window when **Machine-Rotor-Slot** is selected).
Rs	A slot dimension. (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected). Rs is added when the slot type is 3 or 4.

Defining the Rotor Windings and Conductors for a General DC Machine

To define the rotor windings, wires, and conductors:

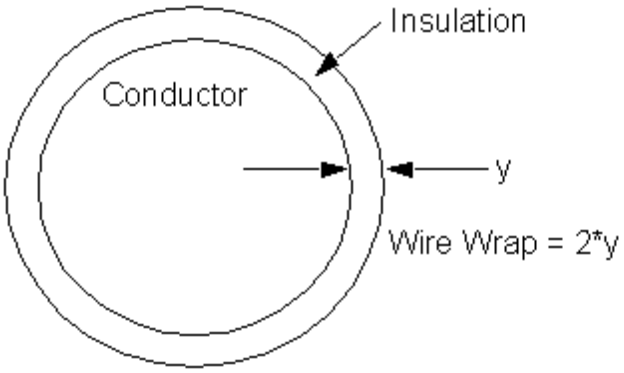
1. To open the **Rotor Slot Winding Properties** window, double-click the **Machine-Rotor-Winding** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Click the **Winding** tab.
3. Select a **Winding Type**:
   a. Click the button for **Winding Type**.
      The **Winding Type** window appears.
   b. Select from one of the following three types of winding:
      - **Lap**
      - **Wave**
      - **Frog Leg**
4. Enter the number of windings in the **Multiplex Number** field (1 for a single winding, 2 for double windings, 3 for triple windings). For a lap winding, the multiplex number is the number of commutators between the start and end of one winding, and the number of parallel branches is equal to the number of poles multiplied by the multiplex number. For a wave winding, the number of parallel branches equals the multiplex number multiplied by two.



5. Enter the number of virtual slots per each real slot in the **Virtual Slots** field. The rotor is assumed to have two layers of conductors, an upper and a lower layer. Each layer of conductors can have a number of windings, which are referred to as virtual slots.

Note	For example, the upper and lower layer can have two windings each, which would yield a virtual slot number of two; for a 12 slot machine, this would yield 24 commutation segments.

6. Enter the total number of conductors in each rotor slot in the **Conductors per Slot** field. This value is the number of turns per coil multiplied by the number of layers. This value is the total number of conductors in one real full rotor slot.
7. Enter the coil pitch, measured in number of slots, in the **Coil Pitch** field. The coil pitch is the number of slots separating one winding. For example, if a coil starts in slot 1 and ends in slot 6, it has a coil pitch of 5.
8. Enter the number of wires per conductor in the **Number of Strands** field. Enter 0 to have RMxprt auto-design this value.
9. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



10. Select the **Wire Size**:
    a. Click the button for **Wire Size**.
    The **Wire Size** window appears.
    b. Select a value from the **Wire Diameter** pull-down list.
    c. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

< ***number*** **>**	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.
USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.
AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.
MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21mm and 2 with a diameter of 0.13mm.

The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>General Options>Machines**command.

    a. When you are done setting the wire size, click **OK** to close the **Wire Size** window and return to the **Properties** window.

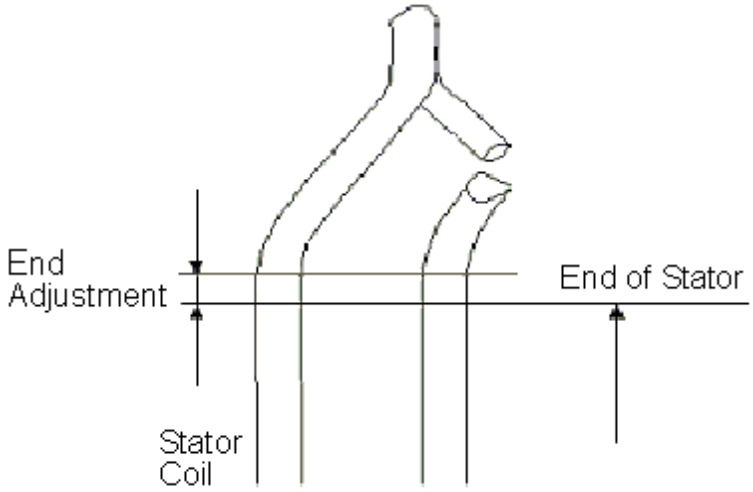
b. Click the **End/Insulation** tab.

c. Select or clear the **Input Half-turn Length** check box.

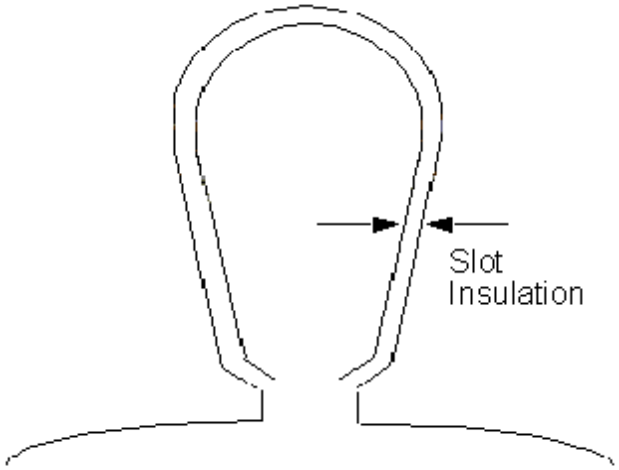
d. Do one of the following:

- If you selected **Input Half-turn Length**, then enter the half-turn length of the armature winding in the Half Turn Length field.
- If you cleared **Input Half-turn Length**, then enter the end length adjustment of the stator coils in the **End Adjustment** field. The end adjustment is the distance one end of the

conductor extends vertically beyond the end of the stator.



5. Enter the inner radius of the base corner in the **Base Inner Radius** field.
6. Enter the inner diameter of the coil tip in the **Tip Inner Diameter** field.
7. Enter the distance between two rotor coils in the **End Clearance** field.
8. Enter the thickness of the slot liner insulation in the **Slot Liner** field.



9. Enter the thickness of the wedge insulation in the **Wedge Thickness** field.
10. Enter the thickness of the insulation layer in the **Layer Insulation** field.
11. Enter the limited slot fill factor for the wire design in the **Limited Fill Factor** field. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.

12. Select the type of equalizer connection from the **Equalizer Connection** pull-down menu. Select from **None**, **Half**, or **Full**.
13. Click **OK** to close the **Properties** window.

Defining Different Size Wires for a General DC Machine Rotor Winding

Use the **Gauge** option if you have a conductor that is made up different size wires.

To define different size wires:

1. In the **Wire Size** window, select **MIXED** from the **Gauge** pull-down menu.
2. Select either **Round** or **Rectangular** as the **Wire Type**.
3. Enter the appropriate wire data in the table:
   - For a round wire:
     - Enter the **Diameter** in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this diameter.
   - For a rectangular wire:
     - Enter the **Width** of the wire in the table.
     - Enter the **Thickness** of the wire in the table.
     - Enter the **Fillet** value in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this data.
4. Click **Add** to add the new wire data.
5. Repeat steps 3 and 4 for each size wire you want to add.
6. When you are finished defining the wires, click **OK** to close the **Wire Size** window and return to the RMxprt **Properties** window.

Note	For example, if one conductor is made up of 5 wires, and 3 of those wires have a diameter of 0.21mm, and the other 2 have a diameter of 0.13mm, then the mixed wire size table will have two lines. The first line will list Diameter = 0.21 and Number = 3. The second line will list Diameter = 0.13 and Number = 2. An equivalent wire diameter is displayed as Wire Size value in the Winding tab in the Properties window.

Rotor Winding Data for General DC Machines

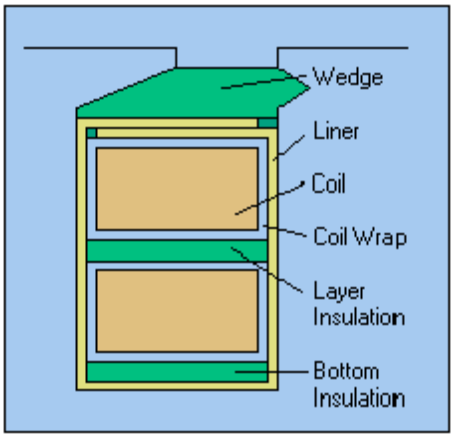
To access the rotor winding data, double-click the **Machine-Rotor-Winding** entry in the project tree.

The **Rotor Winding Data Properties** window contains the following fields:

Winding tab	**Winding Type**	The type of rotor winding. Click the button to open the **Winding Type** window and choose from **Lap**, **Wave**,

		and **Frog Leg**.
	Multiplex Number	Single, double, or triple windings (**1**, **2**, or **3**).
	Virtual Slots	The number of virtual slots per real slot.
	Conductors per Slot	The number of conductors per rotor slot (0 for auto-design).
	Coil Pitch	The coil pitch measured in number of slots.
	Number of Strands	The number of wires per conductor (0 for auto-design).
	Wire Wrap	The thickness of the double-sided wire wrap (0 to automatically obtain this value from the wire library).
	Wire Size	The diameter of the wire (0 for auto-design). Click the button to open the **Wire Size** window where you can specify units, wire type, diameter, and gauge.
End/Insulation tab	**Input Half-turn Length**	Select or clear this check box to specify whether or not you want to enter the half-turn length. When this check box is selected, the **Half Turn Length** field appears the next time you open the **Properties** window. When this check box is selected, the **End Adjustment** field appears instead.
	Half Turn Length	The half-turn length of the armature winding.
	End Adjustment	The end length adjustment of the rotor coils.
	Base Inner Radius	The inner radius of the base corner.
	Tip Inner Diameter	The inner diameter of the coil tip.
	End Clearance	The end clearance between two adjacent coils.
	Slot Liner	The thickness of the slot liner insulation.
	Wedge Thickness	The thickness of the wedge insulation.
	Layer Insulation	The thickness of the insulation layer.
	Limited Fill Factor	The limited slot fill factor for the wire design. The slot fill factor is the ratio

		between the cross-sectional area of all conductors in one slot and the entire slot area.
	Equalizer Connection	The connection type of the equalizer. Select from **None**, **Half**, or **Full**.



Vent Data for General DC Machines

By option, you can insert or remove Vent data for general DC machines. If you have inserted a Vent, the icon appears under the rotor winding in the project tree.

To insert a vent:

1. Right-click on the rotor icon to display the pop-up menu.
2. Click **Insert Vent**.

To remove an existing vent:

1. Right click on the Stator icon to display the pop-up menu.
2. Click **Remove Vent**.

The **Vent Data Properties** window contains the following fields.

Vent Ducts	Number of radial vent ducts
Duct Width	Width of radial vent ducts
Magnetic Spacer Width	Width of magnetic spacer which hold vent ducts. 0 for non-magnetic spacer.
Duct Pitch	Vent ducts
Holes per Row	Number of axial vent holes per row
Inner Hole Diameter	Diameter of vent holes in inner row.

Outer Hole Diameter	Diameter of vent holes in outer row.
Inner Hole Location	Center-to-center diameter of inner row hole vents.
Outer Hole Location	Center-to-center diameter of outer row hole vents.
Banding Slots	Number of axial banding slots to tight the rotor winding.
Width of Banding Slots	Width of axial banding slots
Depth of Banding Slots	Depth of axial banding slots

Defining the Commutator and Brush for a General DC Machine

The commutator allows current transfer between DC terminals or brushes and the rotor coils, providing the current to the system as a function of rotation. Due to the action of the commutator, the corresponding magnetic field has a fixed distribution with respect to the stator.

To define the commutator and brush pairs:

1. To open the **Commutator Data Properties** window, double-click the **Machine>Commutator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Click the **Commutator** tab.
3. Select **Cylinder** or **Pancake Type** as the **Commutator Type**.

Note	When you place the mouse cursor over the commutator type, an outline of the commutator appears.

4. For **Cylinder** commutators, do the following:
   a. Enter the **Commutator Diameter**.
   b. Enter the **Commutator Length**.
5. For **Pancake** commutators, do the following:
   a. Enter the **Outer Diameter**.
   b. Enter the **Inner Diameter**.
6. Enter the thickness of the insulation between two consecutive commutator segments in the **Commutator Insulation** field.
7. Click the **Brush** tab.
8. Enter the **Brush Width**.
9. Enter the **Brush Length**.
10. Enter the number of brush pairs when using a wave armature winding in the **Brush Pairs** field.
11. Enter the angle of displacement from the neutral axis, in mechanical degrees, in the **Brush Displacement** field.

Note	The brush displacement is positive for the counter-clockwise direction. For

	example, if the rotor turns clockwise and the brush displacement is also clockwise, then the angle is negative; if the rotor turns clockwise but the brush displacement is counter-clockwise, then the angle is positive.

12. Enter the voltage drop across one brush pair in the **Brush Drop** field.
13. Enter the mechanical pressure of the brushes as they press against the commutator in the **Brush Press** field.
14. Enter the **Frictional Coefficient** of the brush.

Note	If the **Friction Loss** field is used in the **General** window, the **Brush Press** and **Frictional Coefficient** fields will be hidden in the **Commutator/Brush** window. These fields are shown only when the **Friction Loss** field in the **General** window is set to zero.

15. Click **OK** to close the **Properties** window.

Commutator and Brush Data for General DC Machines

To access the commutator and brush data, double-click the **Machine>Commutator** entry in the project tree.

The **Commutator Data Properties** window contains the following fields:

Commutator tab	**Commutator Type**	The type of commutator. Click the button to open the **Select Commutator Type** window and select from **Cylinder** or **Pancake**.
	Commutator Diameter	For a **Cylinder** commutator type, the diameter of the commutator.
	Commutator Length	For a **Cylinder** commutator type, the length of the commutator.
	Outer Diameter	For a **Pancake** commutator type, the outer diameter of the commutator.
	Inner Diameter	For a **Pancake** commutator type, the inner diameter of the commutator.
	Commutator Insulation	The thickness of the insulation between the two commutator bars.
Brush tab	**Brush Width**	The width of the brush.
	Brush Length	The length of the brush.
	Brush Pairs	The number of brush pairs.
	Brush Displacement	The displacement of the brush from the neutral position, in mechanical degrees (positive for anti-rotating direction).
	Brush Drop	The voltage drop across a one-pair brush.

	Brush Press	The brush press per unit area. (Available only when **Frictional Loss** is set to zero for the machine.)
	Frictional Coefficient	The frictional coefficient of the brush. (Available only when **Frictional Loss** is set to zero for the machine.)

Defining the Shaft Data for a General DC Machine

To define the shaft:

1. To open the **Shaft Data Properties** window, double-click the **Machine>Shaft** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select or clear the **Magnetic Shaft** check box to specify whether or not the shaft is to be made of magnetic material.
3. Select or clear the **No Fan** check box to specify whether or not the machine contains a ventilation fan.
4. If you cleared the **No Fan** check box, then do the following:
   a. Enter the outer diameter of the ventilation fan in the **Fan Diameter** field.
   b. Enter the width of the fan blades in the **Blade Width** field.
5. Click **OK** to close the **Properties** window.

Setting Up Analysis Parameters for a General DC Machine

To define the solution data:

1. To open the **Solution Setup** window, right-click **Analysis** in the project tree, and click **Add Solution Setup**.
2. Click the **General** tab. The **Operation Type** is automatically set to **Motor** for this machine type.
3. Select the **Load Type** used in the motor from the following options:

Const Speed	The speed remains constant in the motor.
Const Power	The output power remains constant in the motor.
Const Torque	The torque remains constant regardless of the speed. In this case, $T_{load} = T_{rated}$, given by the output power divided by the given rated speed.
Linear Torque	The torque increases linearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})$ where T_{rated} is given by the output power divided by the given rated speed.
Fan Load	The load varies nonlinearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})^2$ where T_{rated} is given by the output power divided by the given rated speed.

4. Enter the output power in the **Rated Output Power** field.
5. Enter the applied or output rated DC voltage in the **Rated Voltage** field.

6. Enter the given rated speed in the **Rated Speed** field.
7. Enter the temperature at which the system functions in the **Operating Temperature** field.
8. Click the **DC Machine** tab.
9. Select one of the following from the **Field Exciting Type** pull-down list:
   - **Separately Excited**
   - **Self Excited**
10. Enter the **Exciting Voltage,** and select the units.
11. Enter the **Series Resistance**, and select the units.
12. To automatically obtained the Exciting Voltage and Series Resistance via the **Rated Speed**, rather than entering their values, then select the **Determined by Rated Speed** check box.
13. Click **OK** to close the **Solution Setup** window.

Related Topics

Solution Data for General DC Machines

Solution Data for General DC Machines

To access the solution data, right-click **Analysis** in the project tree, and click **Add Solution Setup**. For this machine type, there is only one tab, the **General** tab.

The **Solution Setup** window contains the following fields:

Operation Type	***General** tab*. Select **Motor** or **Generator** from the pull-down list.
Load Type	*On the* **General** *tab.* For a motor, select from **Const Speed**, **Const Power**, **Const Torque**, **Linear Torque**, and **Fan Load**. The default is **Const Power**. For a generator, select from **Infinite Bus** and **Independent Generato**r.
Rated Output Power	***General** tab*. Type a value for the rated output voltage, and select the units.
Rated Voltage	***General** tab*. Type a value for the rated voltage, and select the units.
Rated Speed	***General** tab*. Type a value for the rated speed, and select the units.
Operating Temperature	***General** tab*. Type a value for the operating temperature, and select the units.
Field Exciting Type	***DC Machine** tab.* Select **Separately Excited** or **Self Excited** from the pull-down list.
Determined by Rated Speed	Select this check box to automatically calculate the **Exciting Voltage** and the **Series Resistance** from the **Rated Speed**, rather than entering the values.

Exciting Voltage	Enter a voltage value in the field, and select the units from the pull-down list.
Series Resistance	Enter a resistance value in the field, and select the units from the pull-down list.

Related Topics

Setting Up Analysis Parameters for a General DC Machine

Claw-Pole Alternators

After you have selected **Claw-Pole Alternators** as your model type, enter the motor data to define the following:

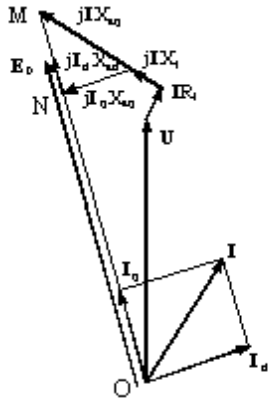
- General data, such as the output power, rated voltage, and speed.
- Stator data.
- Stator slot data.
- Stator winding data.
- Rotor data, such as the slot types and dimensions, rotor diameter, and lamination.
- Rotor pole data.
- Shaft data.
- Optional machine housing.
- Solution data.

Analysis Approach for Claw-Pole Alternators

Claw-pole alternators (or claw-pole synchronous generators) are widely used in auto industry. They receive mechanical energy at the shaft and transform it into electrical energy.

The stator of a claw-pole alternator is equipped with a polyphase winding. The rotor is comprised of claw poles with the same pole number as the stator winding. The claw poles of the rotor are magnetized by a cylinder winding and/or a cylinder permanent magnet. The spinning rotor creates a rotating magnetic field in the air gap, which produces induced voltage in the stator winding.

The performance of a claw-pole alternator is analyzed based on the frequency-domain phasor diagram, as shown in the figure below.



If a claw-pole alternator is equipped with a permanent magnet, the d-axis armature reactance X_{ad} and q-axis armature reactance X_{aq} are about constant. Otherwise, X_{ad} is a linearized nonlinear parameter, and X_{aq} is a linear parameter. The d-axis synchronous reactance X_d and q-axis synchronous reactance X_q are calculated directly from the following:

$$X_d = X_1 + X_{ad}$$

$$X_q = X_1 + X_{aq}$$

Rotor Equipped with an Excitation Winding

If the rotor is equipped with an excitation winding, the exciting current can be adjusted, and the d- and the q-axis currents are obtained based on the following process.

Take the input voltage *U* as the reference phasor, let the power factor angle be *f*, then the current phasor is

The phasor represented by **OM** can be expressed as

$$\mathbf{OM} = \mathbf{U} + \mathbf{I}(R_1 + jX_q)$$

The phasor represented by **OM** can be used to determine the direction of E_0.

θ

Let denote the power angle (the angle that *U* lags E_0), then the angle that *I* lags E_0 is

$$\Psi = \varphi + \theta$$

The d- and the q-axis currents are obtained as follows

$$I_d = I\sin\psi$$

$$I_q = I\cos\psi$$

In the phasor diagrams, the phasor length ON represents the d-axis back emf due to the d-axis resultant flux linkage and is used to determine the d-axis field saturation. From the no-load characteristic curve of the magnetic circuit, E_0, X_{ad} and the excitation current If can be determined based on the frozen method.

Rotor Equipped with a Permanent Magnet Only

If the rotor is equipped with a permanent magnet only, the field excitation can not be adjusted, and the d- and the q-axis currents are obtained based on the following process.

θ

For a given power angle (the angle that *U* lags E_0), we have

$$I_d X_d + I_q R_1 = -(U\cos\theta - E_0)$$

$$-I_d R_1 + I_q X_q = U\sin\theta$$

Solving for I_d and I_q yields.

$$I_d = -\frac{X_q(U\cos\theta - E_0) - R_1 U\sin\theta}{R^2_1 + X_d X_q}$$

$$I_q = \frac{R_1(U\cos\theta - E_0) + X_d U\sin\theta}{R^2_1 + X_d X_q}$$

ω

Let the angle that *I* lags E_0 be , we have

$$\Psi = \tanh\frac{I_d}{I_q}$$

The power factor angle f (the angle that *I* lags *U*) is

$$\varphi = \Psi - \theta$$

Power and Efficiency

The output electric power is

$$P_2 = 3UI\cos\varphi$$

The input mechanical power is

$$P_1 = P_2 + P_{fw} + P_{Cua} + P_{Fe} + P_{Cuf}$$

where P_{fw}, P_{Cua}, P_{Fe}, , and P_{Cuf} are the frictional and wind, the armature copper, the iron-core, the excitation winding copper (if an excitation winding is equipped) losses, respectively.

The input mechanical torque is

$$T_1 = \frac{P_1}{\omega}$$

where ω denotes the synchronous speed in rad/s.

The efficiency of the generator is:

$$\eta = \frac{P_2}{P_1} \times 100\ \%$$

Defining a Claw-Pole Alternator

The general procedure for defining a claw-pole alternator is as follows:

1. Create the alternator project.
2. Double-click the **Machine** entry in the project tree to define the general data.
3. Double-click the **Machine-Stator** entry in the project tree to define the stator geometry.
4. Double-click the **Machine-Stator-Slot** entry in the project tree to define the stator slot dimensions.
5. Double-click the **Machine-Stator-Winding** entry in the project tree to define the stator windings and conductors.
6. Double-click the **Machine-Rotor** entry in the project tree to define the rotor geometry.

7. Double-click the **Machine-Rotor-Pole** entry in the project tree to define the pole, embrace, offset, and air gap data for the rotor pole.
8. Double-click the **Machine-Shaft** entry in the project tree to define the magnetism of the shaft.
9. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
10. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
11. Choose **File>Save** to save the project.
12. Choose **RMxprt>Analyze** to analyze the design.

Note	When you place the cursor over an entry field in the data windows, a brief description of that field appears in the status bar at the bottom of the **RMxprt** window.

Once analyzed, the model can be viewed in the Maxwell 2D Modeler, or it can be used to create a new Maxwell 2D project, and new Maxwell 3D design.

Refer to the *Claw-Pole Alternator Problem* application note, on the technical support page of the Ansys web site, for a specific example of a permanent-magnet DC motor problem.

Defining the General Data for a Claw-Pole Alternator

Use the **General** window to define the basic parameters of the alternator, such as the power settings, speed, and rated voltage.

To define the general data:

1. To open the **General Data Properties** window, double-click the **Machine** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the number of poles for the machine in the **Number of Poles** field. This value is the total number of poles in the stator (or the number of pole pairs multiplied by two).
3. Enter or select the **Number of Phases** (2, 3, or 4).
4. Enter the energy loss due to friction at the given speed in the **Frictional Loss** field.
5. Enter the wind loss due to air resistance measured at the reference speed in the **Wind Loss** field.
6. Enter the given speed in the **Reference Speed** field.
7. Click **OK** to close the **Properties** window.

General Data for Claw-Pole Alternators

To access the general data, double-click the **Machine** entry in the project tree.

The **General Data Properties** window for a three-phase induction motor contains the following fields:

Machine Type	The machine type you selected when inserting a new RMxprt design (Claw-Pole Synchronous Machine).
Frictional Loss	The frictional energy loss (due to friction and air resistance) measured at the reference speed.
Number of Poles	The number of poles the machine contains.
Number of Phases	The number of phases.
Wind Loss	The wind loss measured at the reference speed.
Reference Speed	The given speed of reference.

Defining the Stator Data for a Claw-Pole Alternator

Use the **Stator Properties** windows to define the stator dimensions, slots, windings, and conductors.

The stator is the outer lamination stack where the polyphase voltage windings reside.

To define the general stator data:

1. To open the **Stator Data Properties** window, double-click the **Machine>Stator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the **Outer Diameter** of the stator.
3. Enter the **Inner Diameter** of the stator.
4. Enter the length of the stator core in the **Length** field.
5. Enter the stacking factor for the stator core in the **Stacking Factor** field. This value is a ratio of he effective magnetic length of the core, and ranges from 0 to 1. The stacking factor is defined as the total length minus the total insulation from the laminations, divided by the total length. A value of 1 indicates that the rotor is not laminated.
6. Select a **Steel Type** for the stator core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
7. Enter the **Number of Slots** in the stator.
8. Select the **Slot Type**:
   a. Click the button for the **Slot Type**.
      The **Select Slot Type** window appears.
   b. Select a slot type (available types include 1 through 4).

Note	When you place the mouse cursor over the slot type, a schematic of the selected type appears, displaying the slot dimension variables.

Optionally, check **User Defined Slot** if you wish to define the slot dimensions using the **Slot Editor**.

c. Click **OK** to close the **Select Slot Type** window and return to the **Properties** window.

9. Enter the skew width, measured in slot number, in the **Skew Width** field.
10. Click **OK** to close the **Properties** window.

Stator Data for Claw-Pole Alternators

To access the general stator data, double-click the **Machine>Stator** entry in the project tree.

The **Stator Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the stator core.
Inner Diameter	The inner diameter of the stator core.
Length	The length of the stator core.
Stacking Factor	The stacking factor of the stator core.
Steel Type	The steel type of the stator core. Click the button to open the **Select Definition** window.
Number of Slots	The number of slots the stator core contains.
Slot Type	The type of slots in the stator core. Click the button to open the **Select Slot Type** window.
Skew Width	The skew width measured in slot number.

Defining the Stator Slot Data for a Claw-Pole Alternator

To define the stator slots:

1. To open the **Stator Slot Data Properties** window, double-click the **Machine-Stator-Slot** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Optionally, to automatically design the dimensions of slots **Hs2**, **Bs1**, and **Bs2**, select the **Auto Design** check box.
3. Optionally, to design dimensions of slots **Bs1** and **Bs2** based on the stator tooth width, select the **Parallel Tooth** check box, and enter a value in the **Tooth Width** field.

4. Enter the available slot dimensions.

Hs0	Always available.
Hs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.
Bs0	Always available.
Bs1	Available only when **Auto Design** and **Parallel Tooth** are both cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Bs2	Available only when **Auto Design** and **Parallel Tooth** are both cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Rs	Rs is added when the slot type is 3 or 4.

5. Click **OK** to close the **Properties** window.

Stator Slot Data for Claw-Pole Alternators

To access the stator slot data, double-click the **Machine-Stator-Slot** entry in the project tree.

The **Stator Slot Data Properties** window contains the following fields:

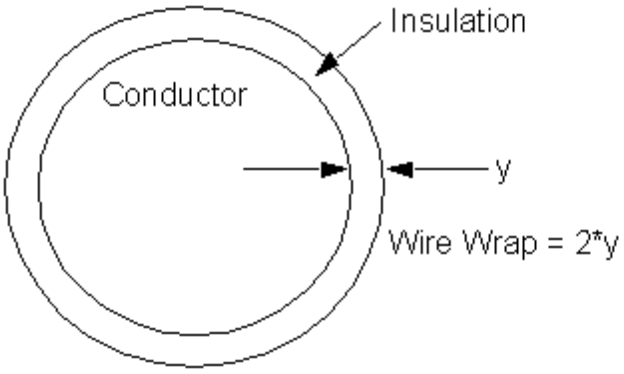
Auto Design	Select or clear this to enable or disable auto-design of slots **Hs2**, **Bs1**, and **Bs2**. When this check box is selected, only two other fields appear in the window: **Hs0** and **Bs0**.
Parallel Tooth	Select this to design **Bs1** and **Bs2** based on the tooth width. When this check box is selected, the **Bs1** and **Bs2** fields are removed, and the **Tooth Width** field is added.
Tooth Width	The tooth width for the parallel tooth, on which **Bs1** and **Bs2** are designed.
Hs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Hs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Hs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs0	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs1	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Bs2	A slot dimension (see the diagram shown in the modeling window when **Machine-Stator-Slot** is selected).
Rs	A slot dimension. (see the diagram shown in the modeling window when **Machine-**

	Stator-Slot is selected). Rs is added when the slot type is 3 or 4.

Defining the Stator Winding Data for a Claw-Pole Alternator

To define the stator windings and conductors:

1. To open the **Stator Slot Winding Properties** window, double-click the **Machine-Stator-Winding** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Click the **Winding** tab.
3. Enter the number of layers in the stator winding in the **Winding Layers** field.
4. Select a **Winding Type**:
   a. Click the button for **Winding Type**.
   The **Winding Type** window appears.
   b. Select from one of the following three types of winding:
      - **Whole Coiled**
      - **Half Coiled**
      - **Editor**
5. Select or enter the number of parallel branches in one phase of the winding in the **Parallel Branches** field.
6. Enter the total number of conductors in each stator slot in the **Conductors per Slot** field. This value is the number of turns per coil multiplied by the number of layers.
7. Enter the coil pitch, measured in number of slots, in the **Coil Pitch** field. The coil pitch is the number of slots separating one winding. For example, if a coil starts in slot 1 and ends in slot 6, it has a coil pitch of 5.
8. Enter the number of wires per conductor in the **Number of Strands** field. Enter 0 to have RMxprt auto-design this value.
9. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



10. Select the **Wire Size**:

a. Click the button for **Wire Size**.
   The **Wire Size** window appears.
b. Select a value from the **Wire Diameter** pull-down list.
c. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

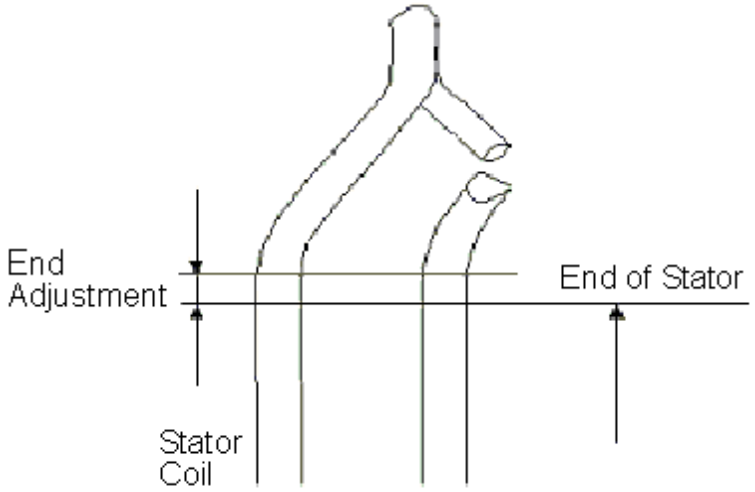
< ***number*** **>**	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.
USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.
AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.
MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21mm and 2 with a diameter of 0.13mm.

The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>General Options>Machines**command.

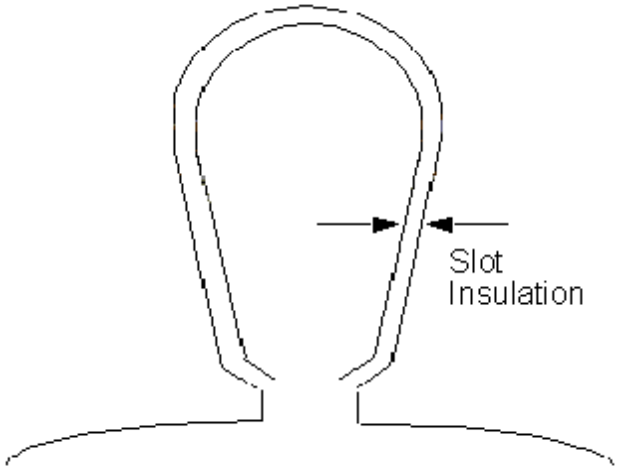
When you are done setting the wire size, click **OK** to close the **Wire Size** window and return to the **Properties** window. Click the **End/Insulation** tab. Select or clear the **Input Half-turn Length** check box. Do one of the following:

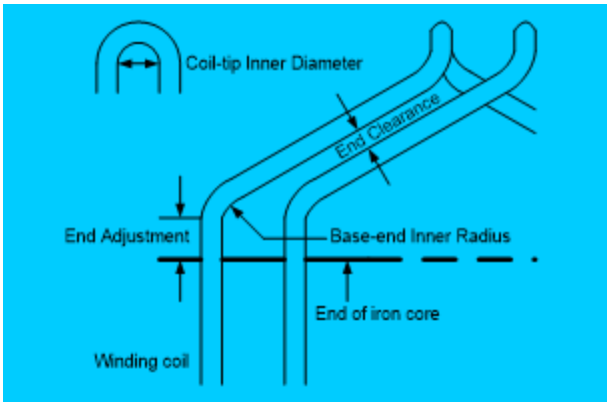
- If you selected **Input Half-turn Length**, then enter the half-turn length of the armature winding in the Half Turn Length field.
- If you cleared **Input Half-turn Length**, then enter the end length adjustment of the stator coils in the **End Adjustment** field. The end adjustment is the distance one end of the

conductor extends vertically beyond the end of the stator.



11. Enter the inner radius of the base corner in the **Base Inner Radius** field.
12. Enter the inner diameter of the coil tip in the **Tip Inner Diameter** field.
13. Enter the distance between two stator coils in the **End Clearance** field.
14. Enter the thickness of the slot liner insulation in the **Slot Liner** field.





15. Enter the thickness of the wedge insulation in the **Wedge Thickness** field.
16. Enter the thickness of the insulation layer in the **Layer Insulation** field.
17. Enter the limited slot fill factor for the wire design in the **Limited Fill Factor** field. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.
18. Click **OK** to close the **Properties** window.

Winding Editor for a Claw-Pole Alternator

For a claw-pole alternator, you may want to specify a different number of conductors for each stator slot. The **Winding Editor** makes this possible by enabling you to specify the number of turns for each coil.

To specify the number of turns for each coil:

1. Click **Machine>Winding>Edit Layout**.
   The **Winding Editor** window appears.
2. In the table in the upper left, set which phase you want for each coil and which slot is the “in” and “out” slot for the current in each coil.
3. If you are working on a quarter or half model, you may want to specify a multiplier by clicking the **Periodic Multiplier** check box and specifying a value.
4. Select or deselect the **Constant Turns** or **Constant Pitch** check boxes, depending on whether you want to be able to change these setting in the table above. When these options are selected, you cannot change the turns or pitch.
5. When you are satisfied with the coil settings, click **OK** to close the **Winding Editor** window.

Defining Different Size Wires for a Claw-Pole Alternator

Use the **Gauge** option in the Wire Size window if you have a conductor that is made up different size wires.

To define different size wires:

1. In the **Wire Size** window, select **MIXED** from the **Gauge** pull-down menu.
2. Select either **Round** or **Rectangular** as the **Wire Type**.
3. Enter the appropriate wire data in the table:
   - For a round wire:
     - Enter the **Diameter** in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this diameter.
   - For a rectangular wire:
     - Enter the **Width** of the wire in the table.
     - Enter the **Thickness** of the wire in the table.
     - Enter the **Fillet** value in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this data.
4. Click **Add** to add the new wire data.
5. Repeat steps 3 and 4 for each size wire you want to add.
6. When you are finished defining the wires, click **OK** to close the **Wire Size** window and return to the RMxprt **Properties** window.

Note	For example, if one conductor is made up of 5 wires, and 3 of those wires have a diameter of 0.21mm, and the other 2 have a diameter of 0.13mm, then the mixed wire size table will have two lines. The first line will list Diameter = 0.21 and Number = 3. The second line will list Diameter = 0.13 and Number = 2. An equivalent wire diameter is displayed as Wire Size value in the Winding tab in the Properties window.

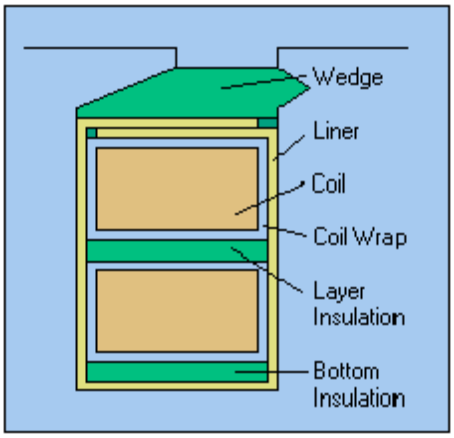
Stator Winding Data for Claw-Pole Alternators

To access the stator winding data, double-click the **Machine-Stator-Winding** entry in the project tree.

The **Stator Winding Data Properties** window contains the following fields:

Winding tab	**Winding Layers**	The number of winding layers.
	Winding Type	The type of stator winding. Click the button to open the **Winding Type** window and choose from **Whole Coiled**, **Half Coiled**, and **Editor**.
	Parallel Branches	The number of parallel branches in the stator winding.
	Conductors per Slot	The number of conductors per stator slot (0 for auto-design).

	Coil Pitch	The coil pitch measured in number of slots.
	Number of Strands	The number of wires per conductor (0 for auto-design).
	Wire Wrap	The thickness of the double-sided wire wrap (0 to automatically obtain this value from the wire library).
	Wire Size	The diameter of the wire (0 for auto-design). Click the button to open the **Wire Size** window where you can specify units, wire type, diameter, and gauge.
End/Insulation tab	**Input Half-turn Length**	Select or clear this check box to specify whether or not you want to enter the half-turn length. When this check box is selected, the **Half Turn Length** field appears the next time you open the **Properties** window. When this check box is selected, the **End Adjustment** field appears instead.
	Half Turn Length	The half-turn length of the armature winding.
	End Adjustment	The end length adjustment of the stator coils, which is the distance one end of the conductor extends vertically beyond the end of the stator.
	Base Inner Radius	The inner radius of the base corner.
	Tip Inner Diameter	The inner diameter of the coil tip.
	End Clearance	The end clearance between two adjacent coils.
	Slot Liner	The thickness of the slot liner insulation.
	Wedge Thickness	The thickness of the wedge insulation.
	Layer Insulation	The thickness of the insulation layer.
	Limited Fill Factor	The limited slot fill factor for the wire design. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.



Defining the Rotor Data for a Claw-Pole Alternator

The rotor is equipped with slots containing copper conductors that are connected to the commutator. The commutator acts as a mechanical rectifier in the motor.

The rotor consists of copper bars in which current is induced by the magnetic fields produced by the stator windings. In the project tree, double-click **Machine-Rotor** and **Machine-Rotor-Pole** to define the rotor and the pole.

To define general rotor data:

1. To open the **Rotor Data Properties** window, double-click the **Machine>Rotor** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the outer diameter of the rotor in the **Outer Diameter** field.
3. Enter the inner diameter of the rotor in the **Inner Diameter** field.
4. Enter the length of the rotor core in the **Length** field.
5. Select a **Steel Type** for the rotor core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
6. Enter the diameter of the rotor yoke in the **Yoke Diameter** field.
7. Click **OK** to close the **Properties** window.

Rotor Data for Claw-Pole Alternators

To access the general rotor data, double-click the **Machine>Rotor** entry in the project tree.

The **Rotor Data Properties** window contains the following fields:

Outer Diameter	The outer diameter of the rotor core.
Inner Diameter	The inner diameter of the rotor core.
Length	The length of the rotor core.
Steel Type	The steel type of the rotor core. Click the button to open the **Select Definition** window.
Yoke Diameter	The diameter of the rotor yoke.

Defining the Rotor Pole for a Claw-Pole Alternator

The rotor pole drives the electromagnetic field which is coupled with the stator windings. Use the **Rotor Pole Data Properties** window to define the rotor pole.

Note	Some of the fields in the **Rotor Pole** window change, or are inactive, depending on the **Rotor Type** you select.

To define the rotor pole:

1. To open the **Rotor Pole Data Properties** window, double-click the **Machine-Rotor-Pole** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the pole embrace at the pole tip in the **Tip Embrace** field. This value must be between 0 and 1, exclusive.
3. Enter the pole embrace at the pole root in the **Root Embrace** field. This value must be between 0 and 2, exclusive.
4. Enter the pole thickness at the pole tip in the **Tip Thickness** field.
5. Enter the pole thickness at the pole root in the **Root Thickness** field.
6. Enter the **Pole Length**.
7. Enter the **Slot Depth**.
8. Enter the **Shoe Thickness**.
9. Select the type of magnet to use in the rotor pole from the **Magnet Type** pull-down menu.
10. If a magnet is being used, enter its length in the **Magnet Length** field.
11. Enter the width of the second air gap in the **Second Air Gap** field.
12. Click **OK** to close the **Properties** window.

Rotor Pole Data for Claw-Pole Alternators

To access the pole rotor data, double-click the **Machine-Rotor-Pole** entry in the project tree.

The **Rotor Pole Data Properties** window contains the following fields:

Tip	The pole embrace at the pole tip. Must be > 0 and < 1.

Embrace	
Root Embrace	The pole embrace at the pole root. Must be > 0 and < 2.
Tip Thickness	The pole thickness at the pole tip.
Root Thickness	The pole thickness at the pole root.
Pole Length	The length of the pole.
Slot Depth	The slot depth.
Shoe Thickness	The shoe thickness.
Magnet Type	The type of magnet. Click the button to open the **Select Definition** window. For all pole types.
Magnet Length	The length of the magnet (if a magnet is used).
Second Air Gap	The width of the second air gap.

Defining the Shaft Data for a Claw-Pole Alternator

To define the shaft:

1. To open the **Shaft Data Properties** window, double-click the **Machine>Shaft** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select or clear the **Magnetic Shaft** check box to specify whether or not the shaft is to be made of magnetic material.
3. Click **OK** to close the **Properties** window.

Shaft Data for Claw-Pole Alternators

To access the shaft data, double-click the **Machine>Shaft** entry in the project tree.

The **Shaft Data Properties** window contains the following fields:

Magnetic Shaft	Select or clear this check box to indicate whether or not the shaft is made of magnetic material.

Setting Up Analysis Parameters for a Claw-Pole Alternator

To define the solution data:

1. To open the Solution Setup window, right-click **Analysis** in the project tree, and click **Add Solution Setup**.
2. Click the **General** tab. The **Operation Type** is automatically set to **General** for this machine type.
3. Select the **Load Type** used in the machine from the following options:

Infinite Bus	When the generator is connected to an infinite bus of the power system, the terminal voltages remain constant and the output power is determined by the output currents.
Independent Generator	The generator supplies electrical power to a load inductor. For an AC generator, the resistance and inductance of the load inductor are determined by the rated output power, the rated voltage, and the rated power factor. For a DC generator, the resistance of the load resistor is determined by the rated output power and the rated voltage.

4. Enter the output power developed at the shaft of the generator in the **Rated Output Power** field.
5. Enter the RMS line-to-line voltage in the **Rated Voltage** field.
6. Enter the desired output speed of the alternator at the load point in the **Rated Speed** field.
7. Enter the temperature at which the system functions in the **Operating Temperature** field.
8. Click the **Claw-Pole Synchronous Machine** tab.
9. Enter a value in the **Rated Power Factor** field.
10. To enter an **Input Exciting Current**, select the check box, enter a value, and select the units.
11. Click **OK** to close the **Solution Setup** window.

Related Topics

Solution Data for Claw-Pole Alternators

Solution Data for Claw-Pole Alternators

To access the solution data, right-click **Analysis** in the project tree, and click **Add Solution Setup**. For this machine type, there is only one tab, the **General** tab.

The **Solution Setup** window contains the following fields:

Operation Type	***General** tab*. Select **Motor** or **Generator** from the pull-down list. **Generator** is automatically selected for this machine type
Load Type	*On the* **General** *tab.* Select from **Infinite Bus** and **Independent Generato**r.
Rated Output Power	***General** tab*. Type a value for the rated output voltage, and select the units.
Rated Voltage	***General** tab*. Type a value for the rated voltage, and select the units.

Rated Speed	***General** tab*. Type a value for the rated speed, and select the units.
Operating Temperature	***General** tab*. Type a value for the operating temperature, and select the units.
Rated Power Factor	***Claw-Pole Synchronous Machine** tab.* Type a value in the field.
Input Exciting Current	Select this check box, enter a value, and select the units. If this check box is cleared, the value will be calculated automatically rather than entered.

Related Topics

Setting Up Analysis Parameters for a Claw-Pole Alternator

Three-Phase Non-Salient Synchronous Machines (NSSM)

After you have selected Three-Phase Non-Salient Synchronous Machine as your model type, enter the data to define the following:

- *General data, such as the number of poles, frictional loss, and reference speed.*
- *Stator pole and winding data, such as its associated pole dimensions, type of steel, and wire definitions.*
- *Rotor data, such as the slot types and dimensions, rotor diameter, laminations, and windings and conductors.*
- *Commutator and brush data, such as the commutator dimensions and brush length.*
- *Shaft data.*
- *Optional machine housing.*
- *Solution data.*

Also see Analysis Approach for the Three-Phase Non-Salient Synchronous Machine

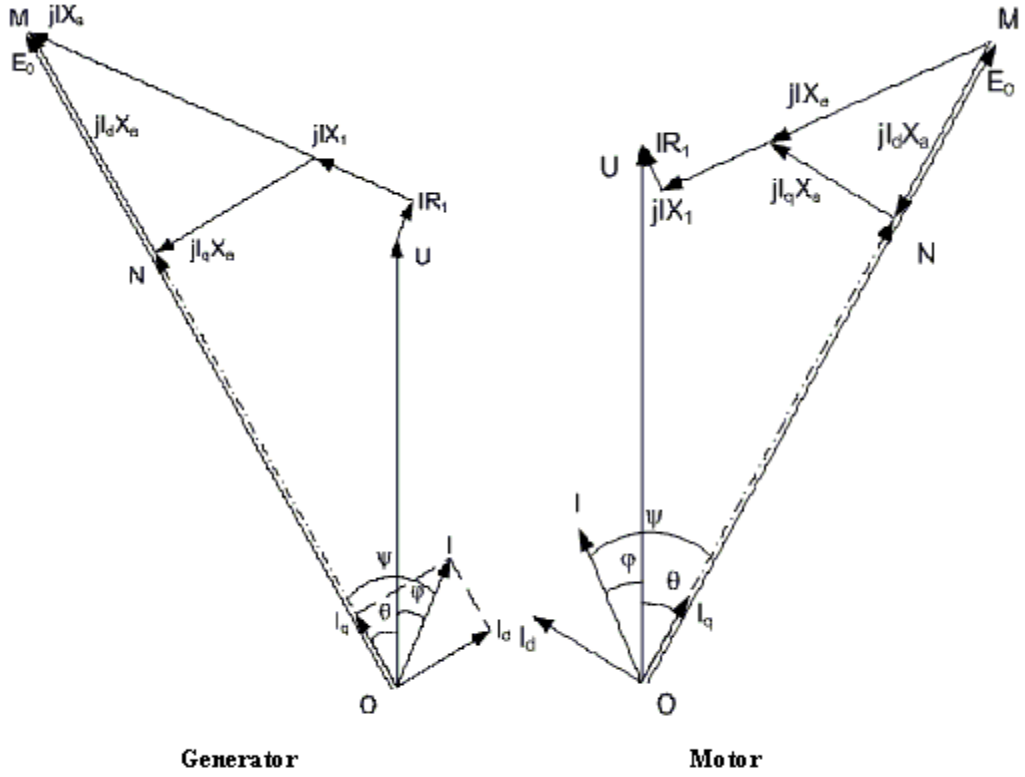
Analysis Approach for Three-Phase Non-Salient Synchronous Machines

The three-phase non-salient-pole synchronous electric machine has two types: the generator and the motor. Their basic structures are the same. The three-phase non-salient-pole synchronous generators are the main source of the electrical energy for industrial, commercial, and private use. They receive the mechanical energy at the shaft and transform it into the electrical energy. The rotor is equipped with a non-salient-pole winding excited by a DC source. The stator is equipped with a three-phase winding that has a sinusoidal spatial distribution. The spinning rotor produces a rotating magnetic field in the air gap of the machine. The frequency of the voltage induced in the stator is given by:

$$f = (pn)/60$$

where *p* is the number of pairs of poles, and n is the mechanical speed of the rotor in rpm, which is called the synchronous speed. The machine is capable of producing both the active and the reactive power as required by the load connected at the stator terminal.

Usually the frequency-domain phasor diagram is adopted to analyze the characteristics. The phasor diagrams for a generator and a motor are shown.



In the figure, R_1, X_1, and X_a are the armature resistance, the armature leakage reactance, and the armature reactance, respectively. In a non-salient-pole synchronous machine, $X_{ad} \cong X_{aq}$ and they are both expressed by X_a. Taking the input voltage **U** as the reference phasor, for a given current:

$$\mathbf{I} = \mathrm{I}\angle -\varphi$$

where φ is the angle **I** lags **U** , which is called the power factor angle.

The internal back EMF induced by the resultant air gap field considering the effects of armature reaction $\mathbf{E_i}$ can be derived from:

$$\boldsymbol{E_i} = \begin{cases} \boldsymbol{U} + (R_1 + \boldsymbol{j}X_1) \cdot \boldsymbol{I} & \textit{for Generator} \\ \boldsymbol{U} - (R_1 + \boldsymbol{j}X_1) \cdot \boldsymbol{I} & \textit{for motor} \end{cases}$$

Based on $\mathbf{E_i}$, the resultant air gap flux considering the effects of armature reaction can be computed, and therefore, the magnetic circuit can be solved. With solved magnetic saturation factor, saturated X_a is derived, and therefore, the no-load induced voltage $\mathbf{E_0}$ with the same magnetic saturation (frozen magnetic circuit) can be calculated from:

$$\boldsymbol{E_0} = \begin{cases} \boldsymbol{E_i} + (\boldsymbol{j}X_a) \cdot \boldsymbol{I} & \textit{for Generator} \\ \boldsymbol{E_i} - (\boldsymbol{j}X_a) \cdot \boldsymbol{I} & \textit{for motor} \end{cases}$$

Let the angle **U** legs $\mathbf{E_0}$ be θ, which is called the power angle for the generator or the torque angle for the motor, then the angle **I** lags $\mathbf{E_0}$ is

$$\psi = \varphi + \theta$$

The d- and the q-axis currents can be obtained respectively as follows:

$$\mathbf{I} = \begin{bmatrix} I_d \\ I_q \end{bmatrix} = I \begin{bmatrix} \sin\psi \\ \cos\psi \end{bmatrix}$$

Based on the magnetic circuit solution and $\mathbf{E_0}$, X_a and the excitation current I_f can be determined based on the frozen method.

1. For the generator:

The output power (electric power) is directly computed from the voltage and the current as:

$$P_2 = 3UI\cos\varphi$$

The input power (mechanical power) is defined as:

$$P_1 = P_2 + P_{fw} + P_{Cua} + P_{Fe} + P_{add} + P_{cuf} + P_{ex}$$

where P_{fw}, P_{Cua}, P_{Fe}, P_{add}, P_{cuf} and P_{ex} are the frictional and wind loss, the armature copper loss, the iron-core loss, the additional loss, the field winding copper loss, and the exciter loss, respectively.

The input mechanical shaft torque is:

$$T_1 = \frac{P_1}{\omega}$$

where ω denotes the synchronous speed in rad/s.

1. For the motor:

The input power (electric power) is directly computed from the voltage and the current as:

$$P_1 = 3UI\cos\varphi$$

The output power (mechanical power) is defined as:

$$P_2 = P_1 - (P_{fw} + P_{Cua} + P_{Fe} + P_{add} + P_{cuf} + P_{ex})$$

where P_{fw}, P_{Cua}, P_{Fe}, P_{add}, P_{cuf} and P_{ex} are the frictional and wind loss, the armature copper loss, the iron-core loss, the additional loss, the field winding copper loss, and the exciter loss, respectively.

The output mechanical shaft torque is:

$$T_2 = \frac{P_2}{\omega}$$

The efficiency is computed for both the generator and the motor by:

$$\eta = \frac{P_2}{P_1} \cdot 100\,\%$$

Related Topics

Defining Three-Phase Non-Salient Synchronous Machines

Defining Three-Phase Non-Salient Synchronous Machines

The general procedure for defining a three-phase non-salient synchronous machine is as follows:

1. Create the non-salient synchronous machine project.
2. After you have selected **Three-Phase Non-Salient Synchronous Machine** as your model type, you must define the following:

- General data, such as number of poles, losses, and reference speed.
- Stator data, such as dimensions, slot type, skew, and laminations.
- Define the Stator slot dimensions.
- Winding data, such as the parallel branches, conductors, and wire dimensions and insulation.
- Rotor data, such as the rotor dimensions, lamination and slot type.
- Define the Rotor slot data.
- Define the Shaft Data.
- Solution data, such as specifying motor or generator application, and rated output voltage and frequency.

You may also use the following options:

- Add a machine housing.
- Add a damper to or remove an existing damper from the rotor;
- Add vents to and remove existing vents from the stator.

Defining the General Data for a Three-Phase NSSM

To access the general data, double-click the **Machine** entry in the project tree.

The **Properties** window for a three-phase non-salient synchronous machine contains the following fields to be entered:

Machine Type	The machine type you selected when inserting a new RMxprt design (Three Phase Non-Salient Synchronous Machine).
Number of	The number of poles the machine contains. This value is the total number of

Poles	poles in the stator (or the number of pole pairs multiplied by two).
Frictional Loss	The frictional energy loss (due to friction) measured at the reference speed.
Windage Loss	The windage loss (due to air resistance) measured at the reference speed.
Reference Speed	The given speed of reference.

Related Topics

Defining the Stator for Three-Phase NSSM

Defining the Stator for Three-Phase NSSM

The stator is the outer lamination stack where the three-phase windings reside.

Double-click the icon **Machine>Stator** in the project tree to display the **Properties** dialog box.

The **Properties** window contains the following fields:

Outer Diameter	The outer diameter of the stator.
Inner Diameter	The inner diameter of the stator.
Length	The length of the stator core.
Stacking Factor	The stacking factor of the stator core.
Steel Type	The steel type of the stator core. Click the button to open the **Select Definition** window.
Number of Slots	The number of slots the stator core contains.
Slot Type	The type of slots in the stator core. Click the button to open the **Select Slot Type** window.
Lamination Sectors	The number of lamination sectors.
Pressboard Thickness	The magnetic press board thickness (enter 0 for a non-magnetic press board).
Skew Width	The skew width measured in slot number.

To define general stator data:

1. To open the **Stator Data Properties** window, double-click the **Machine>Stator** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the **Outer Diameter** of the stator.
3. Enter the **Inner Diameter** of the stator.
4. Enter the length of the stator core in the **Length** field.

5. Enter the stacking factor for the stator core in the **Stacking Factor** field.
6. Select a **Steel Type** for the stator core:
   a. Click the button for **Steel Type**.
      The **Select Definition** window appears.
   b. Select a steel type from the list, or define a new steel type.
   c. Click **OK** to close the **Select Definition** window and return to the **Properties** window.
7. Enter the **Number of Slots** in the stator.
8. Select the **Slot Type**:
   a. Click the button for the **Slot Type**.
      The **Select Slot Type** window appears.
   b. Select a slot type (available types include 1 through 6). Slot types **1** though **4** are filled with round wire. Slot types **5** and **6** are filled with rectangular wire. If **Auto Design** is enabled, the software designs an optimum slot geometry; in this case, you can input the tooth width dimension, and the software determines the slot width accordingly.

Note	When you place the mouse cursor over the slot type, a schematic of the selected type appears, displaying the slot dimension variables.

   Optionally, check **User Defined Slot** if you wish to define the slot dimensions using the **Slot Editor**.

   c. Click **OK** to close the **Select Slot Type** window and return to the **Properties** window.
9. Enter the number of sectors in the **Lamination Sectors** field.
10. Enter the thickness of the magnetic pressboard in the **Pressboard Thickness** field. Enter **0** for a non-magnetic pressboard.
11. Enter the skew width, measured in slot number, in the **Skew Width** field.
12. Click **OK** to close the **Properties** window.

Related Topics

Defining Stator Slots for a Three-Phase NSSM

Defining Stator Slots for a Three-Phase NSSM

To define the slot dimensions:

1. To open the **Stator Slot Data Properties** window, double-click the **Machine>Stator>Slot** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Optionally, to automatically design the dimensions of slots **Hs2**, **Bs1**, and **Bs2**, select the **Auto Design** check box.

3. Enter the available slot dimensions.

Hs0	Always available.
Hs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.
Bs0	Always available.
Bs1	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.
Bs2	Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.

4. Click **OK** to close the **Properties** window.

Related Topics

Defining Stator Windings and Insulation for a Three-Phase NSSM

Defining Stator Windings and Insulation for a Three-Phase NSSM

Double-click the icon **Machine>Stator>Winding** in the project tree to display the **Properties** dialog box, which has two tab sheets: **Winding** and **End/Insulation**.

Define Wires, Conductors and Windings of NSSM Stator

In the **Winding** tab, define the wire, conductor and winding of the stator.

Winding Layers	The number of layers in the stator winding. Select the winding layers from the pull-down list (available choices 1 and 2).
Winding Type	The type of the stator winding. Set the winding type to **Editor** to use the **Winding Editor** dialog to design the coil windings
Parallel Branches	The number of parallel branches in one phase of the stator winding.
Conductors per Slot	The total number of conductors in each stator slot. This value is the number of turns per coil multiplied by the number of layers. Enter 0 to have RMxprt auto-design this value.
Coil Pitch	The coil pitch measured in number of slots. The coil pitch is the number of slots separating one winding. For example, if a coil starts in slot 1 and ends in slot 6, it has a coil pitch of 5.
Number of Strands	The number of wires per conductor. Enter 0 to have RMxprt auto-design this value.
Wire Wrap	The thickness of the double-sided wire wrap. Enter 0 to automatically obtain this value from the wire library.
Wire Size	Wire size (0 for auto-design). You can assign wire size of round wires or rectangle wires. When the slot type you selected is 1 to 4, round wires are used.

	When the slot type you selected is 5 or 6, rectangle wires are used.

Related Topics

Define End Windings and Insulation of NSSM Stator

Winding Editor

Define End Windings and Insulation of NSSM Stator

In the tab sheet **End/Insulation**, define the end winding and the insulation of the stator.

Input Half-turn Length	Select or clear this check box to specify whether or not you want to enter the half-turn length. When this check box is selected, the row **Half Turn Length** appears the next time you open the **Properties** dialog box. When this check box is cleared, the row **End Adjustment** appears instead.
Half-turn Length	The half-turn length of the armature winding. It is available when **Input Half-turn Length** is selected.
End Adjustment	The end length adjustment of the stator coils, which is the distance of one end of the conductor extending vertically beyond the end of the stator. It is available when **Input Half-turn Length** is cleared.
Base Inner Radius	The inner radius of the base corner.
Tip Inner Diameter	The inner diameter of the coil tip.
End Clearance	The end clearance between two adjacent stator coils.
Coil Wrap	Single-side coil wrap insulation thickness.
Slot Liner	The thickness of the slot liner insulation.
Wedge Thickness	The thickness of the wedge insulation.
Limited Fill Factor	The limited slot fill factor for the wire design.
Layer Insulation	The thickness of the insulation layer.
Bottom Insulation	Bottom insulation thickness.

Related Topics

Define Wires, Conductors, and Windings of NSSM Stator

Winding Editor

For a non-salient synchronous motor, you may want to specify a different number of conductors for each stator slot. The **Winding Editor** makes this possible by enabling you to specify the number of turns for each coil. To enable the **Winding Editor**, you must have set the **Winding Property** for the **Winding Type** to **Editor**.

Stator Vent Data for Three-Phase NSSM

To insert a vent on a stator for a three phase synchronous machine:

1. Right click on the stator icon in the project tree to display the shortcut menu.
2. Click **Insert Vent**.

   The vent icon appears in the project tree under the stator.

To remove an existing vent item,

1. Right-click on the stator icon in the project tree to display the shortcut menu.
2. Click **Remove Vent**.

   This removes the vent item from the project tree.

To access the Vent properties for a vent, double click on a vent item. The **Vent Properties** window contains the following fields.

Vent Ducts	The number of radial vent ducts.
Duct Width	The width of the radial vent ducts.
Magnetic spacer width	Width of magnetic spacer which holds vent ducts. O for non-magnetic spacer.
Duct pitch.	Center-to-Center distance between two adjacent Vent ducts

Define NSSM Rotor Data

Double-click the icon **Machine>Rotor** in the project tree to display the **Properties** dialog box, which has one tab sheet: **Rotor**. In the **Rotor** tab, define the rotor general data.

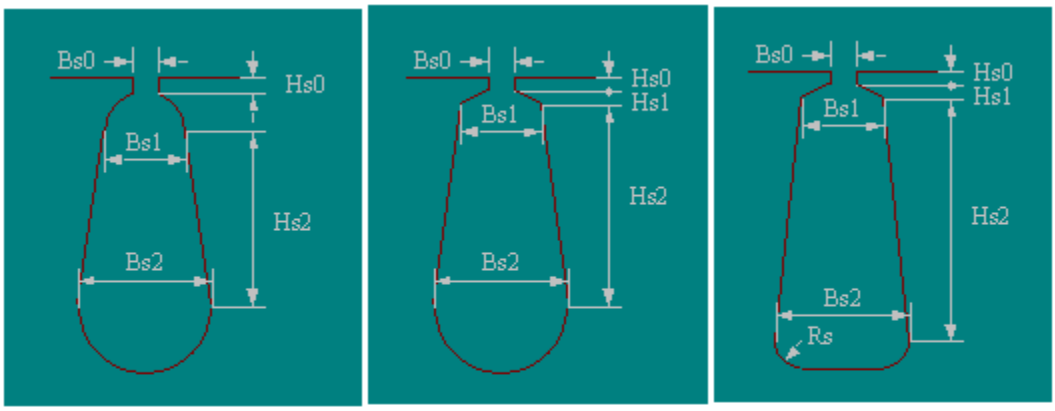
Outer Diameter	The outer diameter of the rotor core.
Inner Diameter	The inner diameter of the rotor core.
Length	The length of the rotor core.
Stacking Factor	Stacking factor of the rotor core.
Steel Type	Select a steel type for the rotor core material.
Press Board Thickness	Magnetic press board thickness, 0 for non-magnetic press board.
Indexing Slots	Number of indexing slots of the rotor core used to determine slot

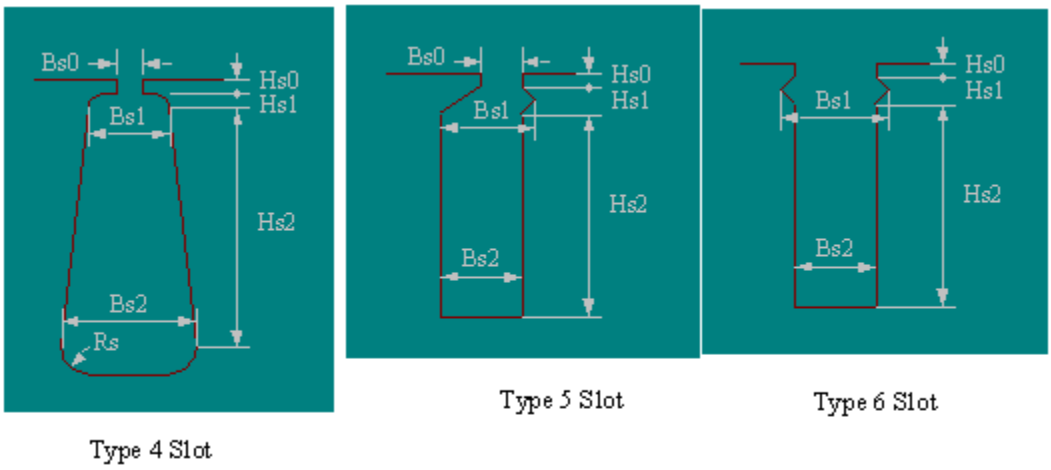
	pitch.
Real Slots	Number of Slots of the rotor core.
Slot Type	Slot type of the rotor core. There are six types of rotor slots.

Define NSSM Rotor Slot

Double-click the icon **Machine>Rotor>Slot** in the project tree to display the **Properties** dialog box.

In the **Slot** tab, define the available rotor slot dimensions as illustrated. There are in total six types of slots that are available:





Related Topics

Define NSSM Rotor Winding

Define NSSM Rotor Winding

The rotor winding is equipped on the rotor pole to provide the excitation for the magnetic field.

Double click the icon **Machine>Rotor>Winding** in the project tree to display the **Properties** dialog box, where you define the wires and physical dimensions of the rotor winding.

In the **Winding** tab, the following are defined:.

Parallel Branches	The number of parallel branches in the rotor winding.
Conductors per Slot	The number of conductors per slot (0 for auto-design).
Number of Strands	The number of wires per conductor (0 for auto-design).
Wire Wrap	The thickness of the double-sided wire wrap (0 to automatically obtain this value from the wire library).
Wire Size	Wire size (0 for auto-design). You can assign wire size of round wires or rectangle wires. When you select Round Wire for Winding Type, round wires are used ((refer to section 8.4.1 Assign Round Wire Sizes). Otherwise, rectangle wires are used (refer to section 8.4.2 Assign Rectangular Wire Size).

In the **End/Insulation** tab the following are defined:

Input Half-turn Length	Select or clear this check box to specify whether or not you want to enter the half-turn length. When this check box is selected, the row **Half Turn Length** appears the next time you open the **Properties** dialog box. When this check box is cleared, the row **End Adjustment** appears instead.
Half-turn Length	The half-turn length of the armature winding. It is available when **Input Half-turn Length** is selected.
End Adjustment	One-side end extended length.
Inner Fillet Radius	Inner fillet radius at the span corner.
End Clearance	End clearance between two adjacent coils.
Coil Wrap	Insulation: single-side coil wrap thickness.
Slot Liner	Insulation: slot liner thickness.
Wedge Thickness	Insulation: wedge thickness.
Bottom Insulation	Insulation: bottom insulation thickness.
Limited Cross Height	The limited cross-section height for the winding design or arrangement, or Overall Height as shown in Figure 12.12 (0 for available maximum area).

Winding Fillet	The size of the winding fillet.

Related Topics

Define NSSM Shaft Data

Rotor Vent Data for NSSMs

By option, you can add vents to a rotor in a three-phase NSSM.

To add a vents to the rotor:

1. Select the rotor icon in the project tree.
2. Right-click to display the pop-up menu and select **Insert Vent**.
   The vent icon appears in the project tree under the rotor.

To remove a vent to stator in a three-phase induction motor.

1. Select the rotor icon in the project tree.
2. Right-click to display the pop-up menu and select **Remove Vent**.
   The vent icon disappears in the project tree under the stator.

The Vent data for the NSSM rotor includes the following fields.

Surface Ducts	Number of surface tangential vent ducts
Surface Duct Width	Width of surface tangential vent ducts
Surface Duct Depth	Depth of surface tangential vent ducts
Surface Duct Pitch	Pitch of surface tangential vent ducts
Axial Ducts	Number of axial vent ducts per pole
Axial Duct Width	Width of axial vent ducts in main teeth
Axial Duct Depth	Depth of axial vent ducts in main teeth

Define NSSM Shaft Data

To define the shaft:

1. Click the icon **Machine>Shaft** in the project tree to display the **Properties** dialog box.
2. In the tab sheet Shaft, select or clear the check box **Magnetic Shaft** to specify whether or not the shaft is to be made of the magnetic material.
3. Click **OK** to close the **Properties** dialog box.

Analysis Setup for Three-Phase Non-Salient Synchronous Machines

Add Solution Setup for NSSM

To set up the solution data:

1. Right click the icon Analysis in the project tree, then click Add Solution Setup from the shortcut menu to display the dialog box Properties. There are two tab sheets.
2. On the **General** tab, define the solution setup data.

Operation Type	Two options from the pull-down list: Generator and Motor.
Load Type	Select a load type for the motor or generator from the pull-down list (refer to section 7.8 Assign Load Types).
Rated Apparent Power	The output electric apparent power in kVA developed at the terminal for the generator, or **Rated Output Power**: The output mechanical power in kW developed at the shaft for the motor.
Rated Voltage	The RMS line-to-line voltage.
Rated Speed	The desired synchronous speed.
Operating Temperature	The temperature at which the system functions, and select the units. The Operating Temperature will affect all winding resistances and therefore affect all ohmic losses.

3. On the **NSSM** tab, define the connection data:

Rated Power Factor	The rated power factor. For generators, the rated output power is determined by the rated apparent power multiplying the rated power factor.
Winding Connection	Select Wye or Delta from the pull-down list.
Exciter Efficiency	The percentage efficiency of the exciter used to supply the rotor winding with the DC current if it is mechanically connected to the shaft of the generator. The efficiency value ranges between 0% and 100% and will only affect the total efficiency result.
Input Exciting Current	If the check box is selected, the companying edit box is enabled. You need to input the exciting current value and select the units if needed.
Exciting Current	Exciting current for rated operation.

4. Click OK to close the pop-up dialog box

Validate NSSM Solution Setup

1. Click **RMxprt>Validation Check** to display the information box Validation Check.
2. If any items do not pass validation, use the diagnostic information in the window to resolve any issues.
3. Click Close to close the information box Validation Check.
4. When the design has been validated, click RMxprt>Analyze All.
5. The analysis progress is shown in the Progress window and the analysis message is shown in the Message Manager.

Design Output for Non-Salient Synchronous Machines

When **RMxprt** has completed a solution, you can display and analyze the results in the following ways:

View Performance

To view the solutions:

Click **RMxprt>Results>Solution Data** to display the information box Solutions. It has three tab sheets.

In the tab sheet Performance, from the pull-down list Data, you have 13 different data tables for the line start permanent magnet motor, which can be used to define Output Variables for design optimization:

- FEA input Data
- Field Winding
- Full-load Magnetic Variables
- Important Factors
- Material Consumption
- No-load Magnetic Variables
- Rated Operation
- Stator Slot
- Stator Winding
- Steady State Parameters per Unit
- Transient Data
- Transient Data per Unit
- Unsaturated Steady State Parameters

View Design Sheet

In the tab sheet Design Sheet, you have 12 sets of information, as follows:

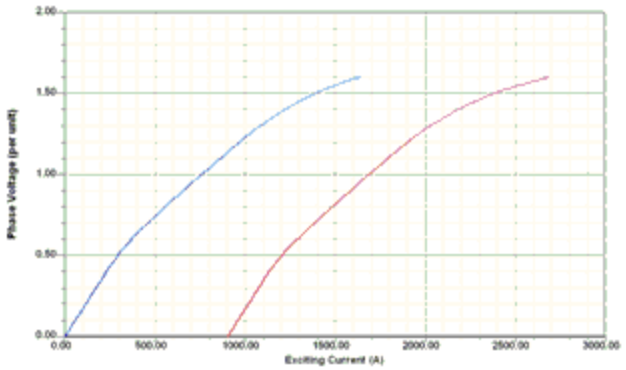
- General Data
- Stator Data
- Stator Winding Data
- Rotor Data
- Field Winding Data
- Some Factors and Material Consumption
- Unsaturated Steady State Parameters
- No Load Magnetic Data
- Full Load Magnetic Data
- Full Load Electric Data
- Transient Parameters and Time Constants
- Transient FEA Input Data

Note	To print the **Design Sheet**: Right click the **Design Sheet**, select **Print** from the shortcut menu, select the printer and other parameters from the dialog box **Print**, and click **OK** to print.

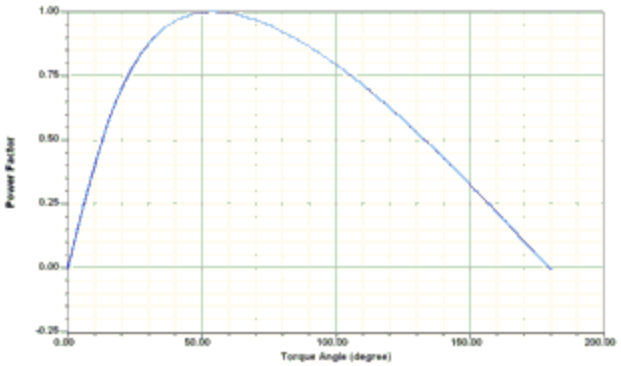
View Curves

In the tab sheet **Curves**, from the pull-down list Name, you have 10 curves as shown:

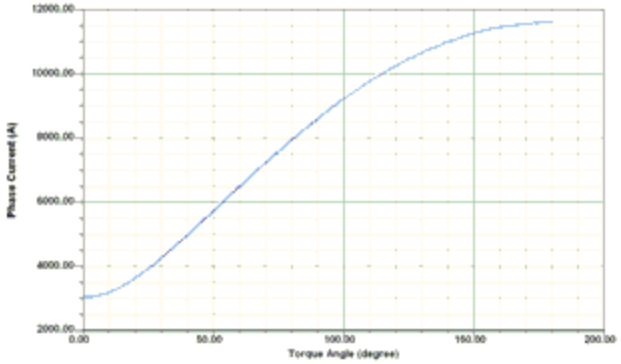
Phase Voltage vs Exciting Current



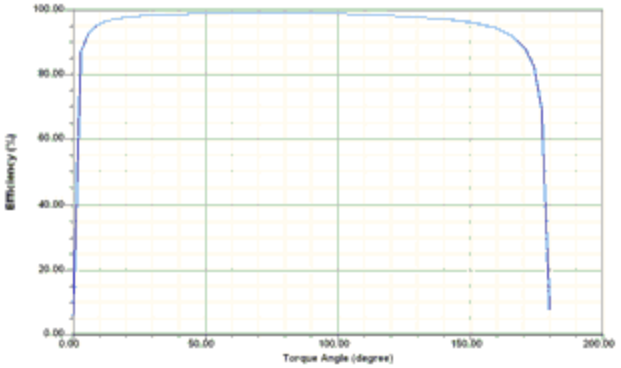
Power Factor vs Torque Angle



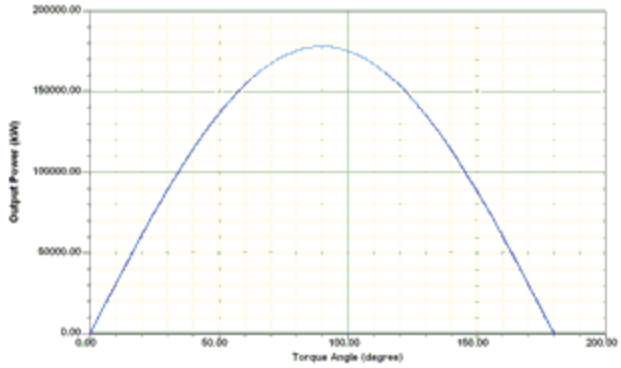
Armature Phase Current vs Torque Angle



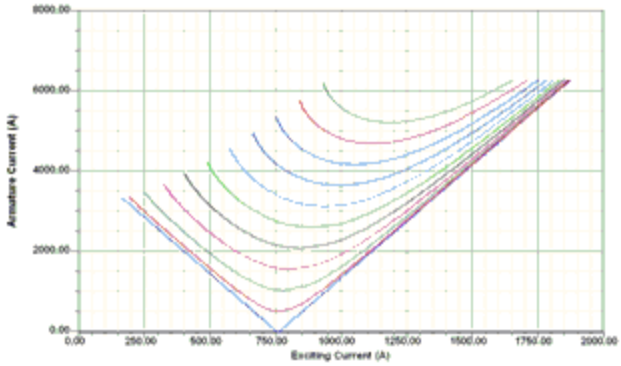
Efficiency vs Torque Angle



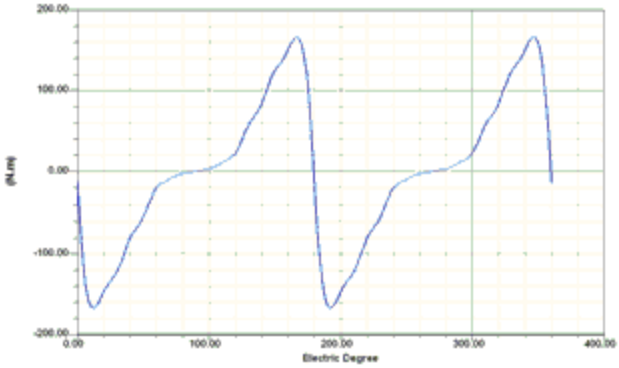
Output Power vs Torque Angle



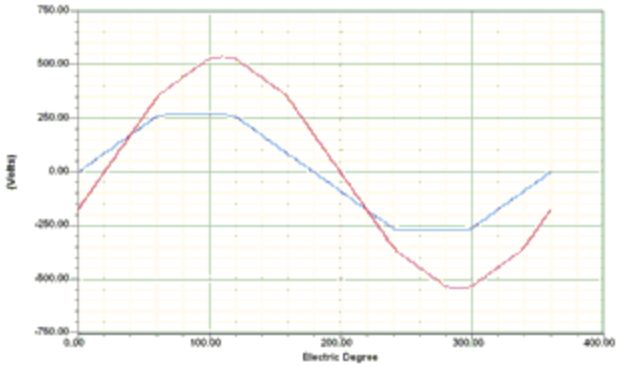
Armature Current vs Exciting Current



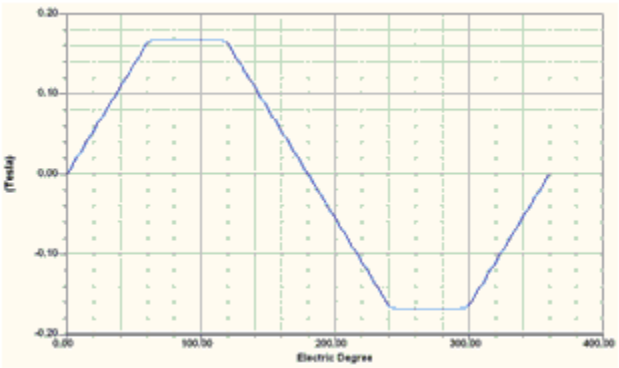
Cogging Torque in Two Teeth



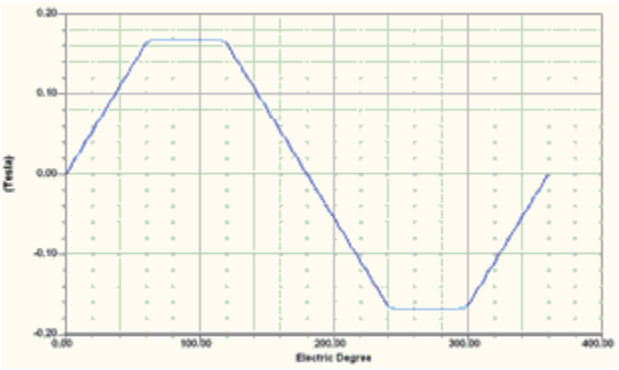
Induced Coil Voltages at No Load



Air-Gap Flux Density at No-Load



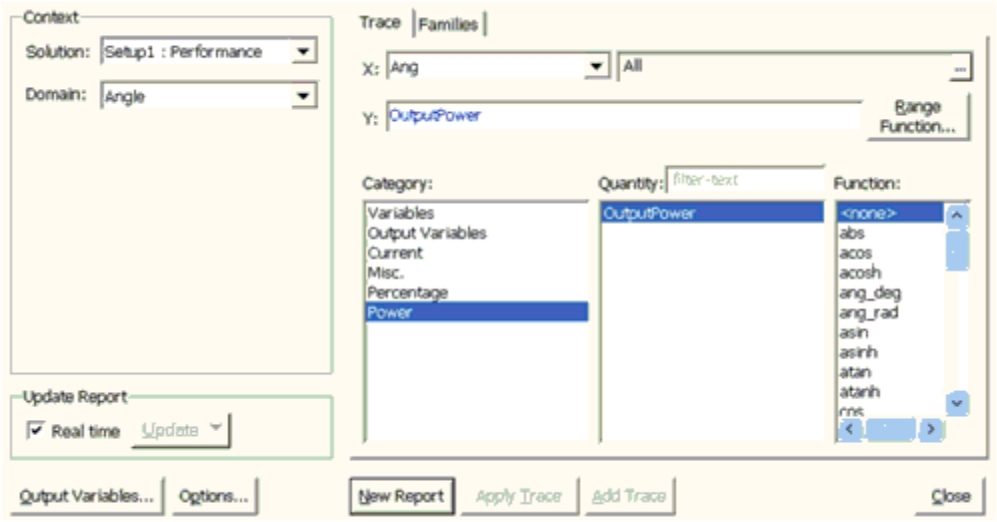
Induced Winding Voltages at No-Load



Note	To print the plots from the **Curve**: Right click on the plot, select **Print** from the shortcut menu, select the printer and other parameters from the dialog box **Print**, and click **OK** to print.

Create Reports

1. Click **RMxprt>Results>Create RMxprt Report>Rectangular Plot**.
   The dialog box **Report** appears as shown:



2. Under the tab sheet **Trace**, there are **Variables, Output Variables, Current, Misc, Percentage, and Power** under the **Category** column. Select one from the **Category** column, select the traces that belong to it from the **Quantity** column, and click the button **Add Trace** to add them one by one. Finally click the button **New report** to create the plot. You can always add additional curves to the same plot by repeating the process.
3. Double click the icon **Results>XY Plot1** to display the graph with multiple traces in a new window.

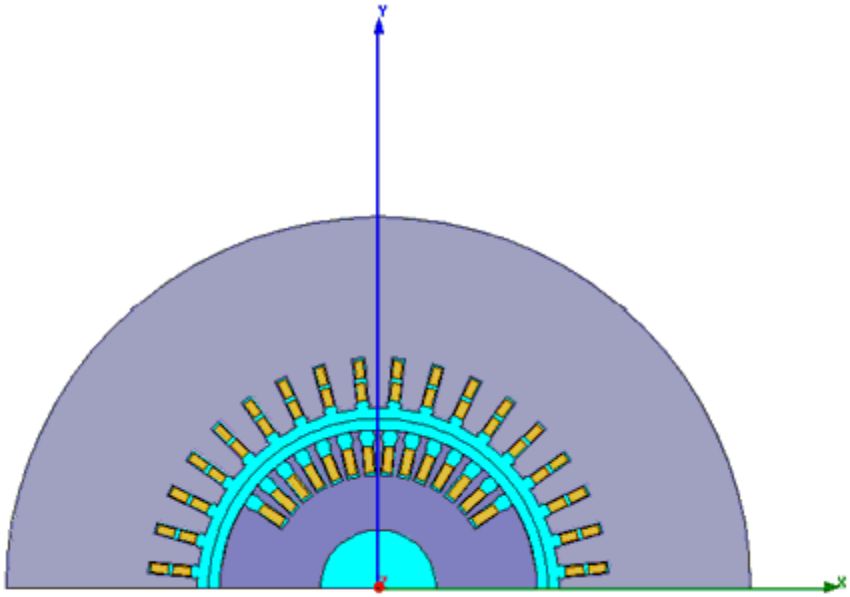
Note	To print the plots from the **Curves**: Right click on the plot, select **Print** from the shortcut menu, select the printer and other parameters from the dialog box **Print**, and click **OK** to print. To get a screen shot of from the **Curves**: Right click on the plot, select **Copy Image**, then paste to a destination file.

Transient FEA of the Non-Salient Synchronous Machines

If you expect to continue the transient or electromagnetic-field FEA with Maxwell2D, you can create a Maxwell2D design directly from RMxprt. For transient FEA, RMxprt can create a Maxwell2D design with all setups completed.

Create Maxwell 2D Design

Click the command **RMxprt>Analysis Setup>Create Maxwell Design…** in RMxprt to create a Maxwell2D design with Auto setup checked (refer to subsection 5.2.1 Create Maxwell 2D Design). A Maxwell2D design called Maxwell2DDesign1 is created with the displayed geometry as shown below. All setups are automatically completed by RMxprt.



Review Maxwell2D Design Setups

This section reviews all setups automatically completed by RMxprt. For detailed setup process, please refer to **APPENDIX Setup Maxwell 2D Designs**.

Solution Type Setup

Click **Maxwell 2D>Solution Type…** in Maxwell2D, you can review that the **Solution Type** is set as **Magnetic Transient**.

Model Setup

1. Model Depth

   Click **Maxwell 2D>Design Setting…** in Maxwell2D and click **Set Model Depth…** tab to review the Model Depth: 3590 mm.
2. Motion Type

   Double click on **Maxwell2DDesign1>Model>MotionSetup1** in the Project Manager window. In the **Type** tab you can review the **Motion Type** being set as **Rotation**, and the **Moving Vector** as **Positive Global: Z**.
3. Initial Position

In the **Data** tab of the **Motion Setup** panel you can review the **Initial Position** being set as **100** deg with **Rotate Limit** unchecked. The rotor initial position is set to such a position that the initial flux linkage of the phase-A winding is at its negative maximum value.

4. Mechanical Load

   In the **Mechanical** tab of the **Motion Setup** panel you can review the **Angular Velocity** being set as **3000** rpm with **Consider Mechanical Transient** unchecked.

5. Symmetry Multiplier

   Right click on **Maxwell2DDesign1>Model** in the Project Manager window, and select **Set Geometry Multiplier** in the pop-up panel, you can review that the **Symmetry Multiplier** is set as **2**.

Boundary Setup

1. Vector Potential Boundary

   Double click on **Maxwell2DDesign1>Boundaries>VectorPotential1** in the Project Manager window, you can review that the highlighted outer half circle in the geometry is set as the **Vector Potential Boundary**, and its value is set as **0**.

2. Independent Boundary

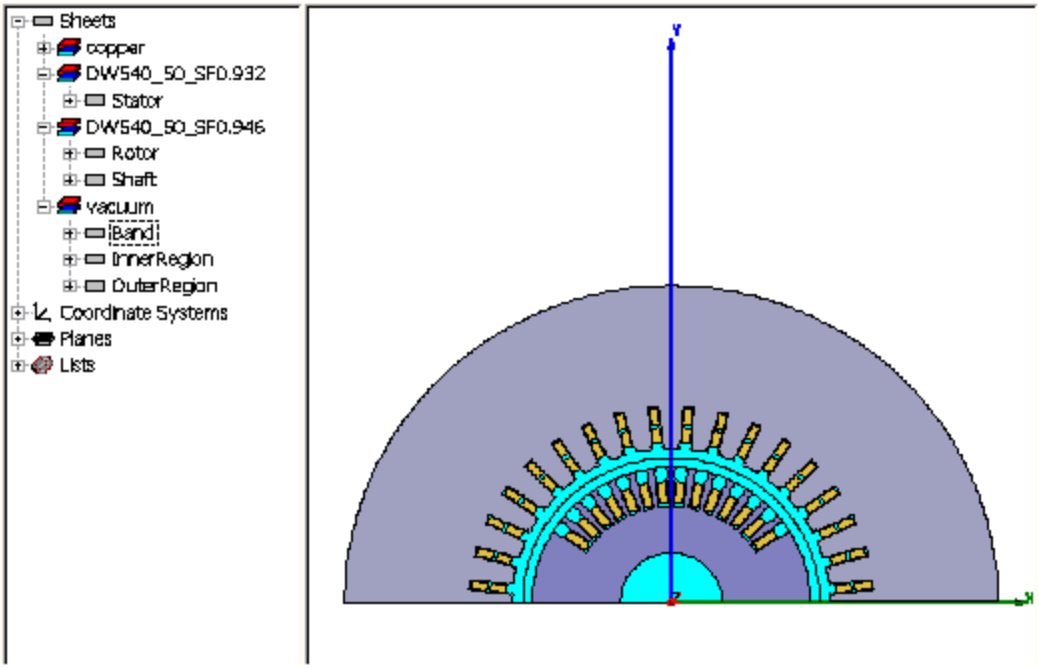
   Click on **Maxwell2DDesign1>Boundaries>Independent1** in the Project Manager window, you can review that the highlighted arrowhead line from left to right in the geometry is set as the **Independent Boundary**.

3. Dependent Boundary

   Double click on **Maxwell2DDesign1>Boundaries>Dependent1** in the Project Manager window, you can review that the highlighted arrowhead line from right to left in the geometry is set as the **Dependent Boundary**, and the relation of the dependent boundary to the independent boundary is set as
   Bs = -Bm. This is because the geometry includes only 1 magnetic pole of the machine.

Material Assignment

In the Maxwell2D modeler windows history tree, you can see that all stator and rotor coil terminals are assigned to material copper by default. **Band**, **InnerRegion** and **OuterRegion** are assigned as vacuum as shown:



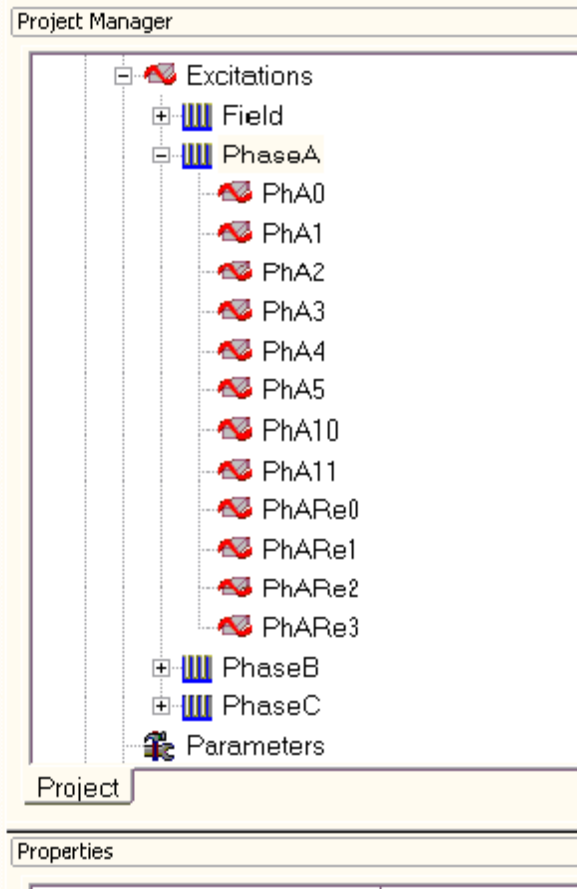
Two new materials called DW540_50_SF0.932, and DW540_50_SF0.946 are automatically created for **Stator** and **Rotor**, based on the original material of DW540_50 used in RMxprt and the equivalent stacking factors of 0.932 and 0.946. Shaft is also assigned as DW540_50_SF0.946, because the shaft is defined as magnetic in RMxprt.

Excitation Setup

1. Windings

   Click on **Maxwell2DDesign1>Excitations>PhaseA** in the Project Manager window, all objects assigned to this phase are highlighted in the modeler window. In the Properties window, you can review all winding properties: **Voltage** for **Winding Type**; **Stranded** for **IsSolid**; **0.00226117** ohms for **Resistance**; **8.87325e-005** H for **Inductance**; **1** for **Number of Parallel Branches**; 11267.7 * sin(2*pi*50*time-43.4944*pi/180) for **Voltage**, where **50** is the **frequency** in Hz, **11267.7** is the **phase peak voltage** in Volts, **pi** is a predefined constant, and **time** is a predefined variable for time. By using sin function instead of cos function, the applied voltage and back EMF are in phase. Therefore, a phase shift in the applied voltage source will be the power angle of the motor. 43.4944 degrees is the power angle at full load operation. The values for resistance, inductance and number of parallel branches are obtained from the TRANSIENT FEA INPUT DATA section in RMxprt design sheet.

   Clicking on **PhaseB**, **PhaseC**, or **Field**, you can review all objects assigned to this winding in the modeler window, and winding properties in the **Properties** window.

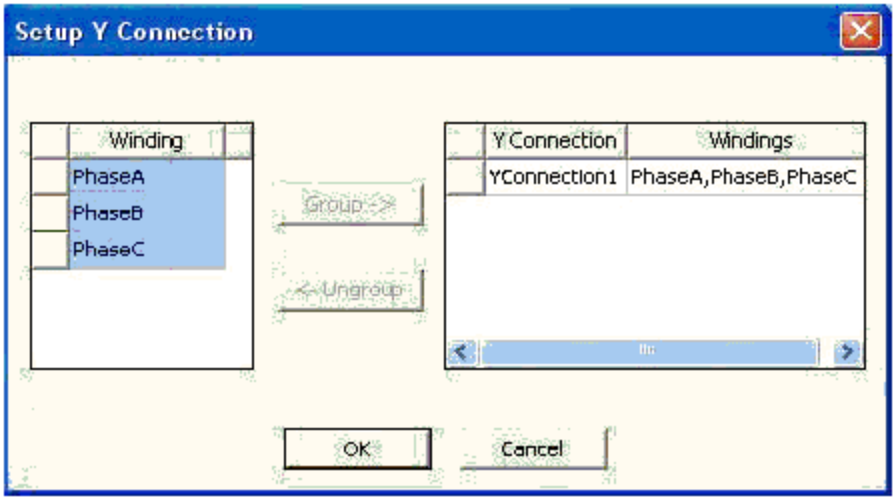


2. Coil Terminals

   A winding consists of several coil terminals, and two coil terminals represent a coil in a complete 2D model. Since we are working with only one-half of the motor structure, one coil terminal can represent one complete coil with independent/dependent boundary conditions provided. A coil terminal has properties of **Number of Conductors** and **Polarity Type**. **Number of Conductors** is the number of turns per coil, and it is equal to the **Number of Turns** given in RMxprt divided by number of coils per phase. **Polarity Type** defines the direction of the current in the coil; it can be either positive or negative. Expand a winding and click on a coil terminal, you can review the object corresponding to this coil terminal in the modeler window and all coil terminal properties in the **Properties** window. In this example, **Number of Conductors** of A, B, and C coil terminals is assigned as 1, and it is 12 for the Field windings. Click on **PhaseB**, **PhaseC**, or **Field**, you can review all objects assigned to this winding in the modeler window, and winding properties in the properties window.

3. Y Connection for Three-Phase Windings

   Right click on **Maxwell2DDesign1>Excitations** in the Project Manager Window, and click **Setup Y Connection…** in the pop-up panel, you can review the Y -connection setup.



Mesh Operation Setup

Maxwell2D mesh maker can create meshes according to predefined mesh operations. A mesh operation defines one or more conditions for some selected objects for mesh maker to create meshes that satisfy the conditions. RMxprt automatically sets up some mesh operations for different machine parts based on geometry sizes. For this example, mesh operations include Length_Coil (set maximum mesh length as 18 mm for all coils), Length_Field (set maximum mesh length as 19 mm for field winding coils), Length_Main (set the maximum mesh length as 135 mm for all other parts), SurfApprox_Main (set the limited Surface Deviation as 1.175 mm and the limited Normal Deviation as 30 deg for all parts with true-surface arcs).

Click on one of the mesh operations under **Maxwell2DDesign1>Mesh** in the Project Manager window, you can review its properties in the Properties window.

Solution Setup

Click on **Maxwell2DDesign1>Analysis>Setup1** in the Project Manager window, you can review its properties in the Properties window: 0.2s for Stop time, that is 10 periods; 0.0002s for Time step with 100 steps per period.

Analyze Maxwell 2D Design

Before analyzing the Maxwell2D design, you may want to Generate Mesh and Plot Mesh. You may also want to create several Quick Reports to display results.

To analyze the Maxwell2D design: right click on **Maxwell2DDesign1>Analysis>Setup1** in the project tree, and click **Analyze**. While the design is being analyzed, you can update one or all result reports and view the reports.

To update all reports: right click on **Maxwell2DDesign1>Results** in the project tree, and select **Update All Reports**.

To update one report: right click on the report under **Maxwell2DDesign1>Results** in the project tree, and select **Update Report**.

To view all traces of a report: when you double click on the report under **Maxwell2DDesign1>Results** in the project tree, the Modeler window changes to the Results window, and all traces (a curve in a report is a trace) of the selected reports are displayed in the Results window.

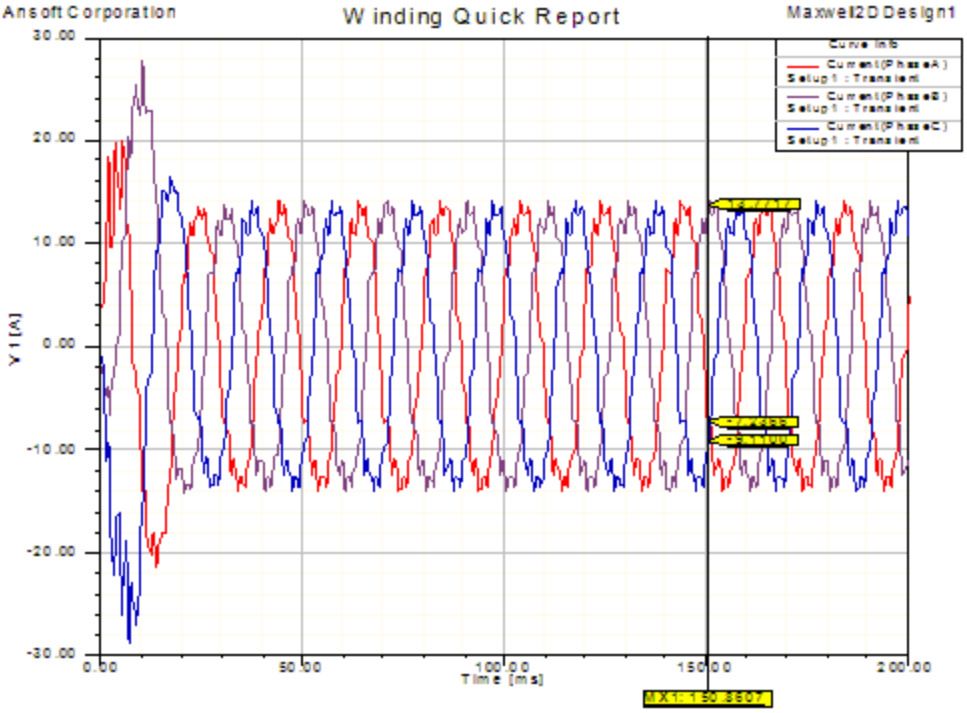
To view a trace of a report: when you click on a trace of a report under **Maxwell2DDesign1>Results** in the project tree, the selected trace is highlighted in the Results window.

To cancel the simulation: right click on the progress bar in the progress window, and pick up **Abort** in the pop-up panel.

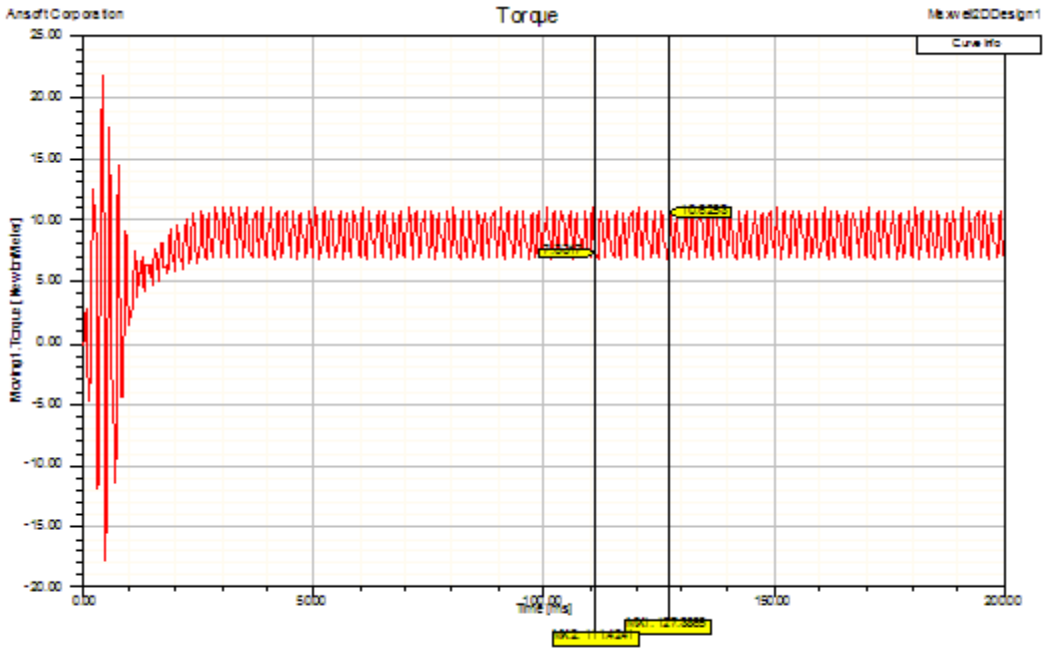
To stop the simulation so that you can continue the simulation later: right click on the progress bar in the progress window, and pick up **Clean Stop** in the pop-up panel.

For this example, the simulated three-phase currents and the electro-magnetic torque are shown in Figure 12.24 and 12.25, respectively.

Right click on the Winding Quick Report in the Results window, and pick up Marker>Add X Marker in the pop-up panel, yellow-shaded boxes are added in the report to indicate X and all Y values. Click on the X box (or the vertical line), and drag it to some place where you see the steady-state peak value of a phase current as shown:



Add X makers in Torque Quick Report to indicate the steady-state maximum and minimum values of torque as shown below. The average torque can be approximately obtained from the maximum and minimum values as Tav = (Tmax + Tmin) / 2 = (10.63 + 7.33) / 2 = 8.98 Nm.



Generic Rotating Machines

The Generic Rotating Machine (GRM) templates allow you to define any of several machine types using the core types listed below.

Core Type	Description
SLOT_AC	Slotted core with AC winding
SLOT_CAGE	Slotted core with squirrel-cage damper
SALNT_POLE	Salient-pole core with field winding
PM_INTERIOR	Interior permanent magnet core
SOLID	Solid core with/without copper/hysteresis sleeve
AXIAL_AC	Axial-flux slotted core with AC winding
AXIAL_CAGE	Axial-flux slotted core with squirrel-cage damper
AXIAL_PM	Axial-flux permanent magnet core
NONS_RELU	Non-salient reluctance core without winding

The machine templates that can be selected, with their associated core type combinations, source types, and position control usage, are listed in the following table.

Note	Refer to "Defining a Generic Rotating Machine" below for detailed information on selecting GRM templates and defining the corresponding machine types.

GRM Machine Type	Stator Type / Rotor Type	Source Type	Position Control
Inner-Rotor Induction Machine	SLOT_AC / SLOT_CAGE	AC	No
Outer-Rotor Induction Machine	SLOT_AC / SLOT_CAGE	AC	No
Solid-Rotor Induction Machine	SLOT_AC / SOLID	AC	No
Wound-Rotor Induction Machine	SLOT_AC / SLOT_AC	AC	No
Double-Fed Induction Generator	SLOT_AC / SLOT_AC	AC	No
Axial-Flux Induction Machine	AXIAL_AC / AXIAL_CAGE	AC	No
Hysteresis Machine	SLOT_AC / SOLID	AC	No
Rotating-Field Synchronous Machine	SLOT_AC / SALNT_POLE	AC	No
Rotating Armature Synchronous Machine	SLOT_AC / SALNT_POLE	AC	No
Reluctance Synchronous Machine	SLOT_AC / NONS_RELU	AC	No
IPM Synchronous Machine	SLOT_AC / PM_INTERIOR	AC	No
Axial-Flux PM Synchronous Machine	AXIAL_AC / AXIAL_PM	AC	No
IPM Brushless DC Motor	SLOT_AC / PM_INTERIOR	DC	Yes
Axial-Flux Brushless DC Motor	AXIAL_AC / AXIAL_PM	DC	Yes

Related Topics

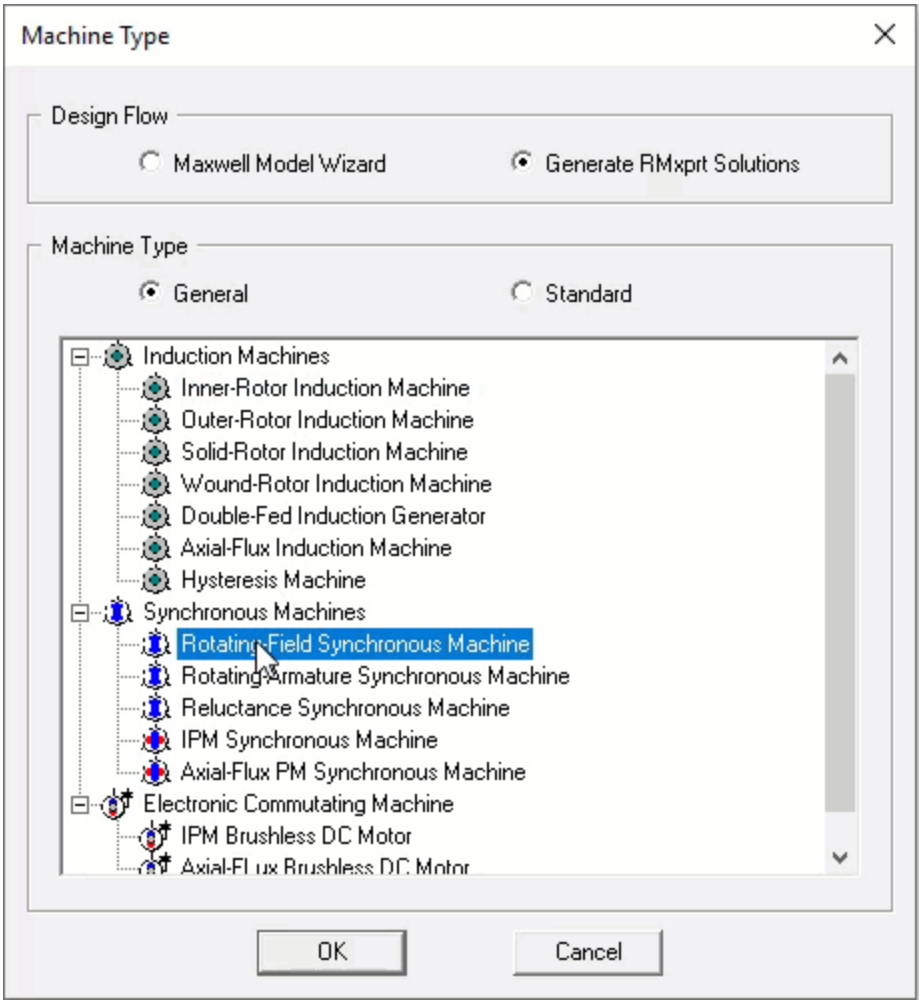
"Defining a Generic Rotating Machine" below

Defining a Generic Rotating Machine

The general procedure for defining a generic rotating machine is as follows:

1. To insert a Generic Rotating Machine into a new or existing project, select **Generate RMxprt Solutions** for the **Design Flow** and **General** for the **Machine Type**.

2. Select the desired machine from the expandable list:



Choose the machine type you wish to define from the following list:

Induction Machines

- Defining an Inner-Rotor Induction Machine
- Defining an Outer-Rotor Induction Machine
- Defining a Solid-Rotor Induction Machine
- Defining a Wound-Rotor Induction Machine
- Defining a Double-Fed Induction Generator
- Defining an Axial-Flux Induction Machine
- Defining a Hysteresis Machine

Synchronous Machines

- Defining a Rotating-Field Synchronous Machine
- Defining a Rotating-Armature Synchronous Machine
- Defining a Reluctance Synchronous Machine
- Defining an IPM Synchronous Machine
- Defining an Axial-Flux PM Synchronous Machine

Electronic Commutating Machine

- Defining an IPM Brushless DC Motor
- Defining an Axial-Flux Brushless DC Motor

Refer to the *Generic Rotating Machine* application note, on the technical support page of the Ansys web site, for a specific example of a problem using a Generic Rotating Machine.

Related Topics

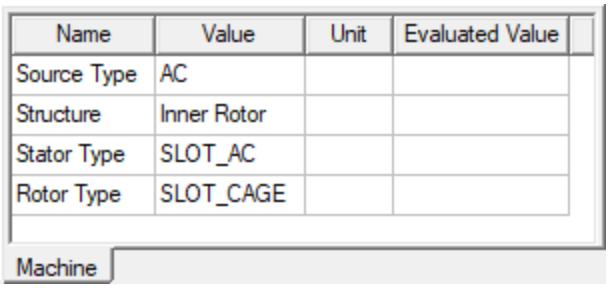
"Generic Rotating Machine Core Types " on page 26-527

Defining an Inner-Rotor Induction Machine

Note	Refer to "Analysis Approach for Three-Phase Induction Motors" on page 26-510 for general information on this machine type.

The general procedure for defining an Inner-Rotor Induction Machine is as follows:

1. Insert an **Inner-Rotor Induction Machine** into a new or existing project by choosing **Generate RMxprt Solutions** for the **Design Flow**, **General** for the **Machine Type**, then select **Inner-Rotor Induction Machine** from the expandable list.
2. Click the **Machine** entry in the project tree to set the machine properties as shown:



3. Click to expand the **Machine>Stator** entry in the project tree to define data for the stator and its associated core and core slot, winding, and optionally, vent for the SLOT_AC core type.
4. Click to expand the **Machine>Rotor** entry in the project tree to define data for the rotor and its associated core, damper and damper slot, and optionally, vent for the SLOT_CAGE core type.

5. Click the **Machine>Shaft** entry in the project tree to define the magnetism, frictional and windage losses, and reference speed of the shaft.
6. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
7. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
8. Choose **File>Save** to save the project.
9. Choose **RMxprt>Analyze** to analyze the design.

Once analyzed, the model can be used to create a new Maxwell 2D or 3D design.

Related Topics

"Analysis Approach for Three-Phase Induction Motors" on page 26-510
"Defining Data for the SLOT_AC Core Type" on page 26-528
"Defining Data for the SLOT_CAGE Core Type" on page 26-530
"Defining the Shaft Data for a Generic Rotating Machine " on page 26-561
"Adding Machine Housings " on page 26-565
"Setting Up Analysis Parameters for a Generic Rotating Machine" on page 26-562
"Saving the Active RMxprt Project" on page 26-4
"Running an RMxprt Simulation " on page 26-142

Analysis Approach for Three-Phase Induction Motors

For a three-phase induction motor, the stator winding (with a sinusoidal spatial distribution and *p* pairs of poles) is connected to a three-phase symmetric voltage power supply. The resulting currents in the stator produce a rotating magnetic field. The rotor winding is often a squirrel cage type with the number of poles dictated by the number of poles in the stator. Currents are induced in the rotor bars and produce, in turn, a second rotating magnetic field. The two rotating fields produce a resultant rotating magnetic field in the air gap of the machine. The interaction of this field in the air gap with the rotor bar currents produces an electromagnetic torque, which acts on the rotor in the direction of the rotation of the field in the air gap. A torque of equal value acts upon the stator in the opposite direction.

The stator winding, which is connected to a phase of the supply system, has *p* coils, each with a symmetric spatial distribution and an opening of $\pi D/2p$, where *D* is the diameter of the winding. In this case, the magnetic field in the air gap has *p* periods, and the winding has *p* pairs of poles.

The performance of three-phase induction motors (IndM3) is analyzed based on the equivalent circuit of one phase in the frequency domain as shown in Figure 1.

In the figure, R1 is the stator resistance, X1 is stator leakage reactance, which consists of stator slot leakage reactance, end-winding leakage reactance, and differential leakage reactance. X2 and R2 are rotor leakage reactance and rotor resistance, respectively. X2 includes rotor slot leakage reactance, end-ring leakage reactance, differential leakage reactance, and skewing leakage reactance. Due to the saturation of the leakage field, X1 and X2 are nonlinear. The

parameters in the equivalent circuit are dependent on the stator and rotor currents. Due to the skin effects, X2 and R2 are the equivalent values from a distributed-parameter circuit, as shown in Figure 2.

They vary with the rotor slip s. All rotor parameters have been referred to the stator side.

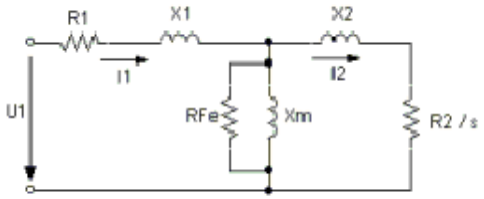


Figure 1

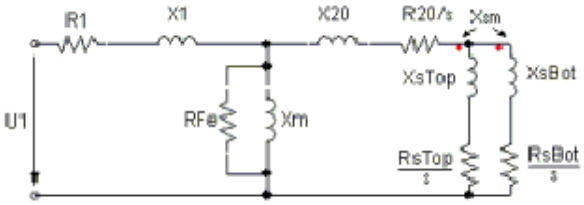


Figure 2

In the exciting branch, Xm is the magnetizing reactance, and RFe is the resistance corresponding to iron-core losses. Xm is a linearized nonlinear parameter that varies with the saturation of the main field.

After a phase voltage U1 is applied to the phase terminals, stator phase current I1 and rotor current I2, which has been referred to the stator, can be easily computed by the circuit analysis. The electromagnetic power Pm, or air-gap power, is computed by the following:

```
Pm = 3 * I2^2 * R2/s
```

The electromagnetic torque Tm is:

$$Tm = Pm / \omega$$

where w is the synchronous speed in rad/s.

The output mechanical shaft torque T2 is:

```
T2 = Tm - Tfw
```

where Tfw is the frictional and wind torque.

The output power is:

$$P2 = T2 * \omega_2$$

where $\omega_2 = \omega * (1 - s)$ and is rotor speed in rad/s.

The input power is:

```
P1 = P2 + Pfw + Pcu2 + PFe + Pcu1 + Ps
```

where Pfw, Pcu2, PFe, Pcu1, and Ps are frictional and wind loss, rotor copper loss, iron-core loss, stator copper loss, and stray loss, respectively.

The power factor is derived from:

```
PF = P1/(m * U1 * I1)
```

The efficiency is computed by:

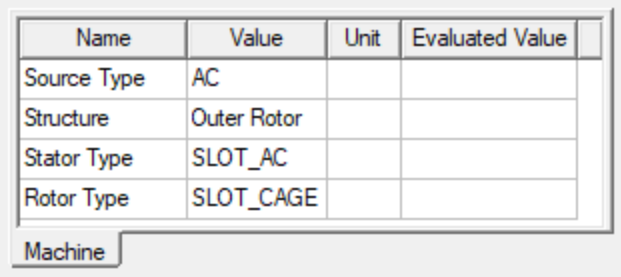
```
eff = P2/P1 * 100%
```

Defining an Outer-Rotor Induction Machine

Note	Refer to "Analysis Approach for Three-Phase Induction Motors" on the next page for general information on this machine type.

The general procedure for defining an Outer-Rotor Induction Machine is as follows:

1. Insert an **Outer-Rotor Induction Machine** into a new or existing project by choosing **Generate RMxprt Solutions** for the **Design Flow**, **General** for the **Machine Type**, then select **Outer-Rotor Induction Machine** from the expandable list.
2. Click the **Machine** entry in the project tree to set the machine properties as shown:



Name	Value	Unit	Evaluated Value
Source Type	AC		
Structure	Outer Rotor		
Stator Type	SLOT_AC		
Rotor Type	SLOT_CAGE		

Machine

3. Click to expand the **Machine>Stator** entry in the project tree to define data for the stator and its associated core and core slot, winding, and optionally, vent for the SLOT_AC core type.
4. Click to expand the **Machine>Rotor** entry in the project tree to define data for the rotor and its associated core, damper and damper slot, and optionally, vent for the SLOT_CAGE core type.
5. Click the **Machine>Shaft** entry in the project tree to define the magnetism, frictional and windage losses, and reference speed of the shaft.
6. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
7. Choose **File>Save** to save the project.
8. Choose **RMxprt>Analyze** to analyze the design.

Once analyzed, the model can be used to create a new Maxwell 2D or 3D design.

Related Topics

"Analysis Approach for Three-Phase Induction Motors" on the next page

"Defining Data for the SLOT_AC Core Type" on page 26-528

"Defining Data for the SLOT_CAGE Core Type" on page 26-530

"Defining the Shaft Data for a Generic Rotating Machine " on page 26-561
"Setting Up Analysis Parameters for a Generic Rotating Machine" on page 26-562
"Saving the Active RMxprt Project" on page 26-4
"Running an RMxprt Simulation " on page 26-142

Analysis Approach for Three-Phase Induction Motors

For a three-phase induction motor, the stator winding (with a sinusoidal spatial distribution and *p* pairs of poles) is connected to a three-phase symmetric voltage power supply. The resulting currents in the stator produce a rotating magnetic field. The rotor winding is often a squirrel cage type with the number of poles dictated by the number of poles in the stator. Currents are induced in the rotor bars and produce, in turn, a second rotating magnetic field. The two rotating fields produce a resultant rotating magnetic field in the air gap of the machine. The interaction of this field in the air gap with the rotor bar currents produces an electromagnetic torque, which acts on the rotor in the direction of the rotation of the field in the air gap. A torque of equal value acts upon the stator in the opposite direction.

The stator winding, which is connected to a phase of the supply system, has *p* coils, each with a symmetric spatial distribution and an opening of $\pi D/2p$, where *D* is the diameter of the winding. In this case, the magnetic field in the air gap has *p* periods, and the winding has *p* pairs of poles.

The performance of three-phase induction motors (IndM3) is analyzed based on the equivalent circuit of one phase in the frequency domain as shown in Figure 1.

In the figure, R1 is the stator resistance, X1 is stator leakage reactance, which consists of stator slot leakage reactance, end-winding leakage reactance, and differential leakage reactance. X2 and R2 are rotor leakage reactance and rotor resistance, respectively. X2 includes rotor slot leakage reactance, end-ring leakage reactance, differential leakage reactance, and skewing leakage reactance. Due to the saturation of the leakage field, X1 and X2 are nonlinear. The parameters in the equivalent circuit are dependent on the stator and rotor currents. Due to the skin effects, X2 and R2 are the equivalent values from a distributed-parameter circuit, as shown in Figure 2.

They vary with the rotor slip s. All rotor parameters have been referred to the stator side.

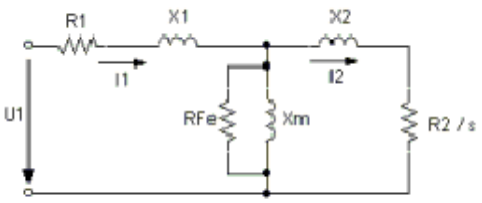


Figure 1

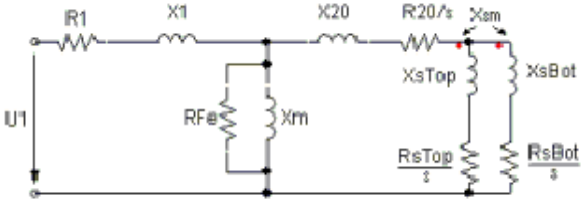


Figure 2

In the exciting branch, Xm is the magnetizing reactance, and RFe is the resistance corresponding to iron-core losses. Xm is a linearized nonlinear parameter that varies with the saturation of the main field.

After a phase voltage U1 is applied to the phase terminals, stator phase current I1 and rotor current I2, which has been referred to the stator, can be easily computed by the circuit analysis. The electromagnetic power Pm, or air-gap power, is computed by the following:

```
Pm = 3 * I2^2 * R2/s
```

The electromagnetic torque Tm is:

$$Tm = Pm/\omega$$

where w is the synchronous speed in rad/s.

The output mechanical shaft torque T2 is:

```
T2 = Tm - Tfw
```

where Tfw is the frictional and wind torque.

The output power is:

$$P2 = T2 * \omega_2$$

where $\omega_2 = \omega * (1 - s)$ and is rotor speed in rad/s.

The input power is:

```
P1 = P2 + Pfw + Pcu2 + PFe + Pcu1 + Ps
```

where Pfw, Pcu2, PFe, Pcu1, and Ps are frictional and wind loss, rotor copper loss, iron-core loss, stator copper loss, and stray loss, respectively.

The power factor is derived from:

```
PF = P1/(m * U1 * I1)
```

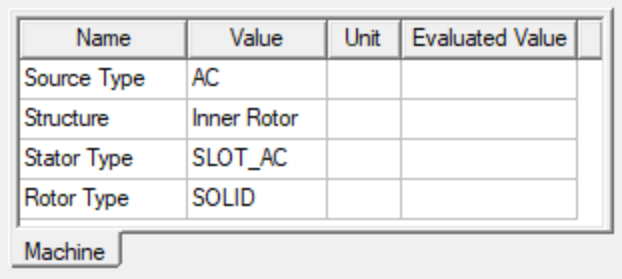
The efficiency is computed by:

```
eff = P2/P1 * 100%
```

Defining a Solid-Rotor Induction Machine

The general procedure for defining a Solid-Rotor Induction Machine is as follows:

1. Insert a **Solid-Rotor Induction Machine** into a new or existing project by choosing **Generate RMxprt Solutions** for the **Design Flow**, **General** for the **Machine Type**, then select **Solid-Rotor Induction Machine** from the expandable list.
2. Click the **Machine** entry in the project tree to set the machine properties as shown:



3. Click to expand the **Machine>Stator** entry in the project tree to define data for the stator and its associated core and core slot, winding, and optionally, vent for the SLOT_AC core type.
4. Click to expand the **Machine>Rotor** entry in the project tree to define data for the rotor core for the SOLID core type.
5. Click the **Machine>Shaft** entry in the project tree to define the magnetism, frictional and windage losses, and reference speed of the shaft.
6. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
7. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
8. Choose **File>Save** to save the project.
9. Choose **RMxprt>Analyze** to analyze the design.

Once analyzed, the model can be used to create a new Maxwell 2D or 3D design.

Related Topics

"Defining Data for the SLOT_AC Core Type" on page 26-528

"Defining Data for the SOLID Core Type" on page 26-531

"Defining the Shaft Data for a Generic Rotating Machine " on page 26-561

"Adding Machine Housings " on page 26-565

"Setting Up Analysis Parameters for a Generic Rotating Machine" on page 26-562

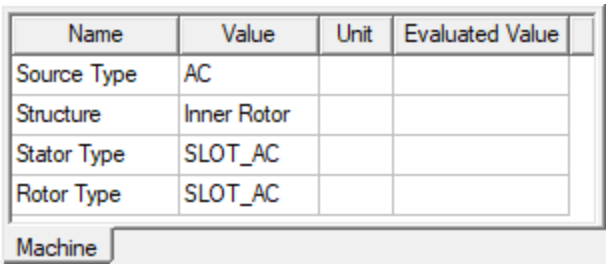
"Saving the Active RMxprt Project" on page 26-4

"Running an RMxprt Simulation " on page 26-142

Defining a Wound-Rotor Induction Machine

The general procedure for defining a Wound-Rotor Induction Machine is as follows:

1. Insert a **Wound-Rotor Induction Machine** into a new or existing project by choosing **Generate RMxprt Solutions** for the **Design Flow**, **General** for the **Machine Type**, then select **Wound-Rotor Induction Machine** from the expandable list.
2. Click the **Machine** entry in the project tree to set the machine properties as shown:



3. Click to expand the **Machine>Stator** entry in the project tree to define data for the stator and its associated core and core slot, winding, and optionally, vent for the SLOT_AC core type.
4. Click to expand the **Machine>Rotor** entry in the project tree to define data for the rotor and its associated core and core slot, winding, and optionally, vent for the SLOT_AC core type.
5. Click the **Machine>Shaft** entry in the project tree to define the magnetism, frictional and windage losses, and reference speed of the shaft.
6. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
7. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
8. Choose **File>Save** to save the project.
9. Choose **RMxprt>Analyze** to analyze the design.

Once analyzed, the model can be used to create a new Maxwell 2D or 3D design.

Related Topics

"Defining Data for the SLOT_AC Core Type" on page 26-528

"Defining the Shaft Data for a Generic Rotating Machine " on page 26-561

"Adding Machine Housings " on page 26-565

"Setting Up Analysis Parameters for a Generic Rotating Machine" on page 26-562

"Saving the Active RMxprt Project" on page 26-4

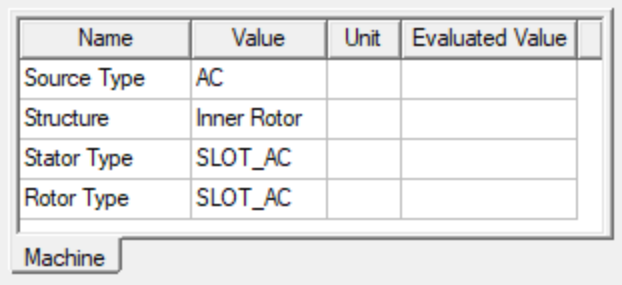
"Running an RMxprt Simulation " on page 26-142

Note	Refer to "Analysis Approach for a Doubly Fed Induction Generator" on the next page for general information on this machine type.

Defining a Double-Fed Induction Generator

The general procedure for defining a Double-Fed Induction Generator is as follows:

1. Insert a **Double-Fed Induction Generator** into a new or existing project by choosing **Generate RMxprt Solutions** for the **Design Flow**, **General** for the **Machine Type**, then select **Double-Fed Induction Generator** from the expandable list.
2. Click the **Machine** entry in the project tree to set the machine properties as shown:



3. Click to expand the **Machine>Stator** entry in the project tree to define data for the stator and its associated core and core slot, winding, and optionally, vent for the SLOT_AC core type.
4. Click to expand the **Machine>Rotor** entry in the project tree to define data for the rotor and its associated core and core slot, winding, and optionally, vent for the SLOT_AC core type.
5. Click the **Machine>Shaft** entry in the project tree to define the magnetism, frictional and windage losses, and reference speed of the shaft.
6. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
7. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data, where **Operation Type** is set to **Wind Generator**.
8. Choose **File>Save** to save the project.
9. Choose **RMxprt>Analyze** to analyze the design.

Once analyzed, the model can be used to create a new Maxwell 2D or 3D design.

Related Topics

"Analysis Approach for a Doubly Fed Induction Generator" below

"Defining Data for the SLOT_AC Core Type" on page 26-528

"Defining the Shaft Data for a Generic Rotating Machine " on page 26-561

"Adding Machine Housings " on page 26-565

"Setting Up Analysis Parameters for a Generic Rotating Machine" on page 26-562

"Saving the Active RMxprt Project" on page 26-4

"Running an RMxprt Simulation " on page 26-142

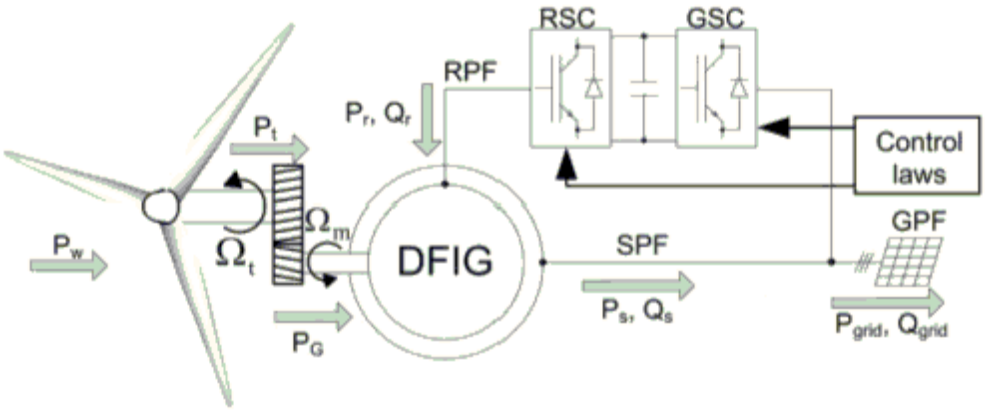
Analysis Approach **for a Doubly Fed Induction Generator**

A Doubly Fed Induction Generator (DFIG) can be configured to operate as any of the following types:

- Generator

DFIG Operating as a Generator

Doubly-fed induction generators (DFIGs) are widely used in wind power systems. A DFIG works as a component of a wind power system, as shown below, where the wind turbine transforms wind energy into mechanical energy, and the DFIG transforms mechanical energy into electrical energy.



For a DFIG, both the stator and the rotor are equipped with poly-phase AC windings. The stator and rotor windings may, or may not, have the same number of phases, but they must have the same number of poles *p*. In order to produce terminal voltages with desired frequency *f* in the stator winding, the rotor winding must be excited by balanced poly-phase currents with the slip frequency sf via an AC-DC-AC convert. Slip *s* is defined as:

$$s = 1 - n/n_0$$

where *n* is the rotor speed, and n_0 is the synchronous speed as given below:

$$n_0 = 60f/p$$

When the rotor speed is *lower* than the synchronous speed, the rotor currents have the same phase sequence as the stator currents, and the rotor winding gets power from the converter. However, when the rotor speed is *higher* than the synchronous speed, the phase sequence of the rotor currents is different from that of the stator currents, and the rotor winding outputs power to the converter.

For a given wind turbine, the power coefficient (the ratio of turbine power to the wind power), is a function of the tip speed ratio (the ratio of the blade tip speed to the wind speed). In order to track the maximum power point, the tip speed ratio must keep constant - at its optimal value. The input mechanical power with Maximum Power Point Tracking (MPPT) must satisfy:

$$P_{mech} = P_{m\_ref} \cdot (\omega_m / \omega_{ref})^3$$

where P_{m_ref} is the turbine power with MPPT at a reference speed of ω_{ref} based on the optimal tip speed ratio, and ω_m is the rotor speed in rad/s.

The rotor mechanical loss is:

$$P_f = P_{f\_ref} \cdot (\omega_m / \omega_{ref})^3$$

where P_{f_ref} is mechanical loss measured at a reference speed of ω_{ref}.

The electro-magnetic power in the air gap is:

$$P_{em} = (P_{mech} - P_f) / (1 - s)$$

Therefore, the stator output electrical power at rated operation is:

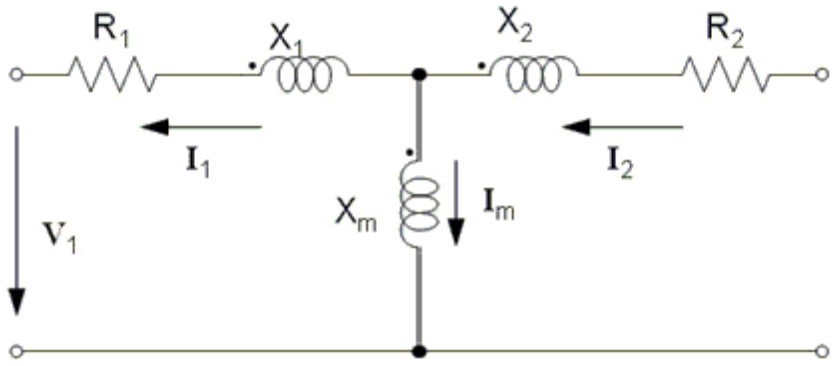
$$P_1 = P_{em} - m_1 I_1^2 R_1 = m_1 V_1 I_1 \cos\varphi$$

where m_1 is the number of phases of the stator winding, R_1 is the stator phase resistance, V_1 is the stator rated phase voltage, I_1 is the rated stator phase current to be determined, and $\cos\varphi$ is the rated power factor. Solving for I_1, one obtains:

$$I_1 = \frac{2P_{em} / m_1}{V_1 \cos\varphi + \sqrt{(V_1 \cos\varphi)^2 + 4R_1 P_{em} / m_1}}$$

Then, based on the equivalent circuit shown below, one obtains:

$$\mathbf{V}_1 = V_1 \angle 0$$
$$\mathbf{I}_1 = I_1 \angle -\varphi$$
$$\mathbf{E}_m = \mathbf{V}_1 + \mathbf{I}_1 (R_1 + jX_1)$$
$$\mathbf{I}_m = (\mathbf{E}_m / X_m(E_m))$$
$$\mathbf{I}_2 = I_2 \angle -\varphi_2 = \mathbf{I}_1 + \mathbf{I}_m$$



Now, rotor input electrical power can be computed as:

$$P_2 = sP_{em} + m_2 I_2^{\ 2} R_2$$

where m_2 is the number of phases of the rotor winding.
The electromagnetic torque T_{em} is:

$$T_{em} = \frac{P_{em}}{\omega}$$

where ω ? denotes the synchronous speed in rad/s.
The input mechanical torque on the shaft is:

$$T_{mech} = T_{em} + T_f$$

where T_f denotes the frictional torque.
The total electrical output power is:

$$P_{elec} = P_1 - P_2 - p_{Fe}$$

where p_{Fe} is the core loss.

The efficiency is defined as:

$$\eta = \frac{P_{elec}}{P_{mech}} \times 100\%$$

Defining an Axial-Flux Induction Machine

Note	Refer to "Analysis Approach for Three-Phase Induction Motors" on page 26-510 for general information on this machine type.

The general procedure for defining an Axial-Flux Induction Machine is as follows:

1. Insert an **Axial-Flux Induction Machine** into a new or existing project by choosing **Generate RMxprt Solutions** for the **Design Flow**, **General** for the **Machine Type**, then select **Axial-Flux Induction Machine** from the expandable list.
2. Click the **Machine** entry in the project tree to set the machine properties as shown:

Name	Value	Unit	Evaluated Value
Source Type	AC		
Structure	Axial-Flux Rotor		
Stator Type	AXIAL_AC		
Rotor Type	AXIAL_CAGE		
Double-Sided	None		
Air Gap Length	0	mm	0mm

Machine

3. Click to expand the **Machine>Stator** entry in the project tree to define data for the stator and its associated core and core slot, and winding for the AXIAL_AC core type.
4. Click to expand the **Machine>Rotor** entry in the project tree to define data for the rotor and its associated core, damper and damper slot for the AXIAL_CAGE core type.
5. Click the **Machine>Shaft** entry in the project tree to define the magnetism, frictional and windage losses, and reference speed of the shaft.
6. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
7. Choose **File>Save** to save the project.
8. Choose **RMxprt>Analyze** to analyze the design.

Once analyzed, the model can be used to create a new Maxwell 2D or 3D design.

Related Topics

"Defining Data for the AXIAL_AC Core Type" on page 26-532

"Defining Data for the AXIAL_CAGE Core Type" on page 26-534

"Defining the Shaft Data for a Generic Rotating Machine " on page 26-561

"Setting Up Analysis Parameters for a Generic Rotating Machine" on page 26-562

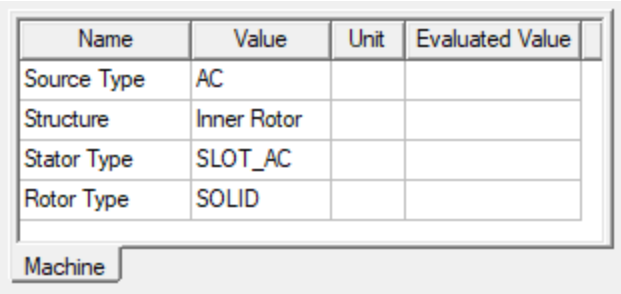
"Saving the Active RMxprt Project" on page 26-4

"Running an RMxprt Simulation " on page 26-142

Defining a Hysteresis Machine

The general procedure for defining a Hysteresis Machine is as follows:

1. Insert a **Hysteresis Machine** into a new or existing project by choosing **Generate RMxprt Solutions** for the **Design Flow**, **General** for the **Machine Type**, then select **Hysteresis Machine** from the expandable list.
2. Click the **Machine** entry in the project tree to set the machine properties as shown:



3. Click to expand the **Machine>Stator** entry in the project tree to define data for the stator and its associated core and core slot, winding, and optionally, vent for the SLOT_AC core type.
4. Click to expand the **Machine>Rotor** entry in the project tree to define data for the rotor core for the SOLID core type, where **Sleeve Type** is set to **Hysteresis**.
5. Click the **Machine>Shaft** entry in the project tree to define the magnetism, frictional and windage losses, and reference speed of the shaft.
6. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
7. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
8. Choose **File>Save** to save the project.
9. Choose **RMxprt>Analyze** to analyze the design.

Once analyzed, the model can be used to create a new Maxwell 2D or 3D design.

Related Topics

"Defining Data for the SLOT_AC Core Type" on page 26-528

"Defining Data for the SOLID Core Type" on page 26-531

"Defining the Shaft Data for a Generic Rotating Machine " on page 26-561

"Adding Machine Housings " on page 26-565

"Setting Up Analysis Parameters for a Generic Rotating Machine" on page 26-562

"Saving the Active RMxprt Project" on page 26-4

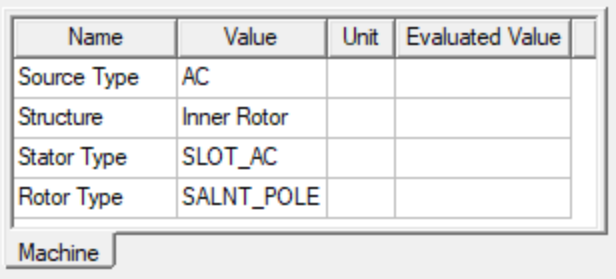
"Running an RMxprt Simulation " on page 26-142

Defining a Rotating-Field Synchronous Machine

Note	Refer to "Analysis Approach for Three-Phase Synchronous Machines" on page 26-323 for general information on this machine type.

The general procedure for defining a Rotating-Field Synchronous Machine is as follows:

1. Insert a **Rotating-Field Synchronous Machine** into a new or existing project by choosing **Generate RMxprt Solutions** for the **Design Flow**, **General** for the **Machine Type**, then select **Rotating-Field Synchronous Machine** from the expandable list.
2. Click the **Machine** entry in the project tree to set the machine properties as shown:



3. Click to expand the **Machine>Stator** entry in the project tree to define data for the stator and its associated core and core slot, winding, and optionally, vent for the SLOT_AC core type.
4. Click to expand the **Machine>Rotor** entry in the project tree to define data for the rotor and its associated core, pole, optional brush, and optional damper and damper slot for the SALNT_POLE core type.
5. Click the **Machine>Shaft** entry in the project tree to define the magnetism, frictional and windage losses, and reference speed of the shaft.
6. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
7. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.

8. Choose **File>Save** to save the project.
9. Choose **RMxprt>Analyze** to analyze the design.

Once analyzed, the model can be used to create a new Maxwell 2D or 3D design.

Related Topics

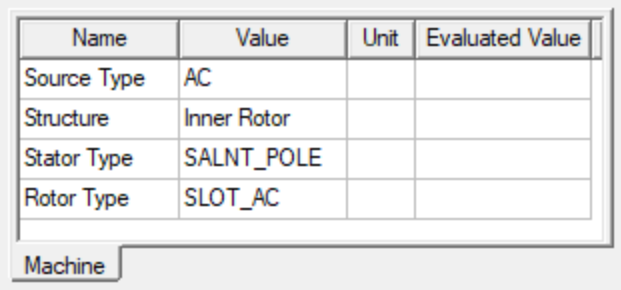
"Defining Data for the SLOT_AC Core Type" on page 26-528
"Defining Data for the SALNT_POLE Core Type" on page 26-536
"Defining the Shaft Data for a Generic Rotating Machine " on page 26-561
"Adding Machine Housings " on page 26-565
"Setting Up Analysis Parameters for a Generic Rotating Machine" on page 26-562
"Saving the Active RMxprt Project" on page 26-4
"Running an RMxprt Simulation " on page 26-142

Defining a Rotating-ArmatureSynchronous Machine

Note	Refer to "Analysis Approach for Three-Phase Synchronous Machines" on page 26-323 for general information on this machine type.

The general procedure for defining a Rotating-Armature Synchronous Machine is as follows:

1. Insert a **Rotating-Armature Synchronous Machine** into a new or existing project by choosing **Generate RMxprt Solutions** for the **Design Flow**, **General** for the **Machine Type**, then select **Rotating-Armature Synchronous Machine** from the expandable list.
2. Click the **Machine** entry in the project tree to set the machine properties as shown:



3. Click to expand the **Machine>Stator** entry in the project tree to define data for the stator and its associated core, pole, and optional damper and damper slot for the SALNT_POLE core type.
4. Click to expand the **Machine>Rotor** entry in the project tree to define data for the stator and its associated core and core slot, winding, and optionally, vent for the SLOT_AC core type.
5. Click the **Machine>Shaft** entry in the project tree to define the magnetism, frictional and windage losses, and reference speed of the shaft.
6. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.

7. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
8. Choose **File>Save** to save the project.
9. Choose **RMxprt>Analyze** to analyze the design.

Once analyzed, the model can be used to create a new Maxwell 2D or 3D design.

Related Topics

"Defining Data for the SALNT_POLE Core Type" on page 26-536

"Defining Data for the SLOT_AC Core Type" on page 26-528

"Defining the Shaft Data for a Generic Rotating Machine " on page 26-561

"Adding Machine Housings " on page 26-565

"Setting Up Analysis Parameters for a Generic Rotating Machine" on page 26-562

"Saving the Active RMxprt Project" on page 26-4

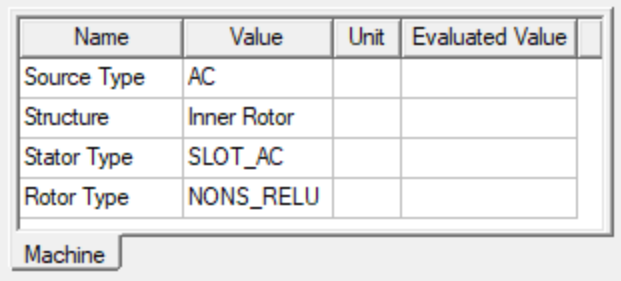
"Running an RMxprt Simulation " on page 26-142

Defining a Reluctance Synchronous Machine

Note	Refer to "Analysis Approach for Three-Phase Synchronous Machines" on page 26-323 for general information on this machine type.

The general procedure for defining a Reluctance Synchronous Machine is as follows:

1. Insert a **Reluctance Synchronous Machine** into a new or existing project by choosing **Generate RMxprt Solutions** for the **Design Flow**, **General** for the **Machine Type**, then select **Reluctance Synchronous Machine** from the expandable list.
2. Click the **Machine** entry in the project tree to set the machine properties as shown:



3. Click to expand the **Machine>Stator** entry in the project tree to define data for the stator and its associated core and core slot, winding, and optionally, vent for the SLOT_AC core type.
4. Click to expand the **Machine>Rotor** entry in the project tree to define data for the rotor and its associated core and pole for the NONS_RELU core type.
5. Click the **Machine>Shaft** entry in the project tree to define the magnetism, frictional and windage losses, and reference speed of the shaft.

6. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
7. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
8. Choose **File>Save** to save the project.
9. Choose **RMxprt>Analyze** to analyze the design.

Once analyzed, the model can be used to create a new Maxwell 2D or 3D design.

Related Topics

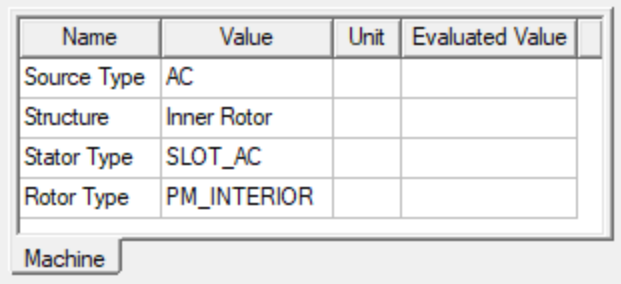
"Defining Data for the SLOT_AC Core Type" on page 26-528
"Defining Data for the NONS_RELU Core Type" on page 26-542
"Defining the Shaft Data for a Generic Rotating Machine " on page 26-561
"Adding Machine Housings " on page 26-565
"Setting Up Analysis Parameters for a Generic Rotating Machine" on page 26-562
"Saving the Active RMxprt Project" on page 26-4
"Running an RMxprt Simulation " on page 26-142

Defining an IPM Synchronous Machine

Note	Refer to "Analysis Approach for Three-Phase Synchronous Machines" on page 26-323 for general information on this machine type.

The general procedure for defining an IPM Synchronous Machine is as follows:

1. Insert an **IPM Synchronous Machine** into a new or existing project by choosing **Generate RMxprt Solutions** for the **Design Flow**, **General** for the **Machine Type**, then select **IPM Synchronous Machine** from the expandable list.
2. Click the **Machine** entry in the project tree to set the machine properties as shown:



3. Click to expand the **Machine>Stator** entry in the project tree to define data for the stator and its associated core and core slot, winding, and optionally, vent for the SLOT_AC core type.
4. Click to expand the **Machine>Rotor** entry in the project tree to define data for the rotor and its associated core and pole for the PM_INTERIOR core type.

5. Click the **Machine>Shaft** entry in the project tree to define the magnetism, frictional and windage losses, and reference speed of the shaft.
6. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
7. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
8. Choose **File>Save** to save the project.
9. Choose **RMxprt>Analyze** to analyze the design.

Once analyzed, the model can be used to create a new Maxwell 2D or 3D design.

Related Topics

"Defining Data for the SLOT_AC Core Type" on page 26-528
"Defining Data for the PM_INTERIOR Core Type" on page 26-545
"Defining the Shaft Data for a Generic Rotating Machine " on page 26-561
"Adding Machine Housings " on page 26-565
"Setting Up Analysis Parameters for a Generic Rotating Machine" on page 26-562
"Saving the Active RMxprt Project" on page 26-4
"Running an RMxprt Simulation " on page 26-142

Defining an Axial-Flux PM Synchronous Machine

Note	Refer to "Analysis Approach for Three-Phase Synchronous Machines" on page 26-323 for general information on this machine type.

The general procedure for defining an Axial-Flux PM Synchronous Machine is as follows:

1. Insert an **Axial-Flux PM Synchronous Machine** into a new or existing project by choosing **Generate RMxprt Solutions** for the **Design Flow**, **General** for the **Machine Type**, then select **Axial-Flux PM Synchronous Machine** from the expandable list.
2. Click the **Machine** entry in the project tree to set the machine properties as shown:

Name	Value	Unit	Evaluated Value
Source Type	AC		
Structure	Axial-Flux Rotor		
Stator Type	AXIAL_AC		
Rotor Type	AXIAL_PM		
Double-Sided	None		
Air Gap Length	0	mm	0mm

Machine

3. Click to expand the **Machine>Stator** entry in the project tree to define data for the stator and its associated core and core slot, and winding for the AXIAL_AC core type.
4. Click to expand the **Machine>Rotor** entry in the project tree to define data for the rotor and its associated core, damper and damper slot for the AXIAL_PM core type.
5. Click the **Machine>Shaft** entry in the project tree to define the magnetism, frictional and windage losses, and reference speed of the shaft.
6. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
7. Choose **File>Save** to save the project.
8. Choose **RMxprt>Analyze** to analyze the design.

Once analyzed, the model can be used to create a new Maxwell 2D or 3D design.

Defining an IPM Brushless DC Motor

Note	Refer to "Analysis Approach for Brushless PMDC Motors" on page 26-351 for general information on this machine type.

The general procedure for defining an IPM Brushless DC Motor is as follows:

1. Insert an **IPM Brushless DC Motor** into a new or existing project by choosing **Generate RMxprt Solutions** for the **Design Flow**, **General** for the **Machine Type**, then select **IPM Brushless DC Motor** from the expandable list.
2. Click the **Machine** entry in the project tree to set the machine properties as shown:

Name	Value	Unit	Evaluated Value
Source Type	DC		
Structure	Inner Rotor		
Stator Type	SLOT_AC		
Rotor Type	PM_INTERIOR		

Machine

3. Click to expand the **Machine>Stator** entry in the project tree to define data for the stator and its associated core and core slot, winding, circuit, and optionally, vent for the SLOT_AC core type.
4. Click to expand the **Machine>Rotor** entry in the project tree to define data for the rotor and its associated core and pole for the PM_INTERIOR core type.
5. Click the **Machine>Shaft** entry in the project tree to define the magnetism, frictional and windage losses, and reference speed of the shaft.
6. Optionally, right-click the **Machine** entry in the project tree and click **Insert Housing** to add a machine housing.
7. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.

8. Choose **File>Save** to save the project.
9. Choose **RMxprt>Analyze** to analyze the design.

Once analyzed, the model can be used to create a new Maxwell 2D or 3D design.

Related Topics

"Analysis Approach for Brushless PMDC Motors" on page 26-351
"Defining Data for the SLOT_AC Core Type" on page 26-528
"Defining Data for the PM_INTERIOR Core Type" on page 26-545
"Defining the Shaft Data for a Generic Rotating Machine " on page 26-561
"Adding Machine Housings " on page 26-565
"Setting Up Analysis Parameters for a Generic Rotating Machine" on page 26-562
"Saving the Active RMxprt Project" on page 26-4
"Running an RMxprt Simulation " on page 26-142

Defining an Axial-Flux Brushless DC Motor

Note	Refer to "Analysis Approach for Brushless PMDC Motors" on page 26-351 for general information on this machine type.

The general procedure for defining an Axial-Flux Brushless DC Motor is as follows:

1. Insert an **Axial-Flux Brushless DC Motor** into a new or existing project by choosing **Generate RMxprt Solutions** for the **Design Flow**, **General** for the **Machine Type**, then select **Axial-Flux Brushless DC Motor** from the expandable list.
2. Click the **Machine** entry in the project tree to set the machine properties as shown:

Name	Value	Unit	Evaluated Value
Source Type	DC		
Structure	Axial-Flux Rotor		
Stator Type	AXIAL_AC		
Rotor Type	AXIAL_PM		
Double-Sided	Stator		
Air Gap Length	0	mm	0mm

Machine

3. Click to expand the **Machine>Stator** entry in the project tree to define data for the stator and its associated core and core slot, and winding for the AXIAL_AC core type.
4. Click to expand the **Machine>Rotor** entry in the project tree to define data for the rotor and its associated core, damper and damper slot for the AXIAL_PM core type.

5. Click the **Machine>Shaft** entry in the project tree to define the magnetism, frictional and windage losses, and reference speed of the shaft.
6. Right-click **Analysis** in the project tree, and click **Add Solution Setup** to define the solution data.
7. Choose **File>Save** to save the project.
8. Choose **RMxprt>Analyze** to analyze the design.

Once analyzed, the model can be used to create a new Maxwell 2D or 3D design.

Related Topics

"Analysis Approach for Brushless PMDC Motors" on page 26-351
"Defining Data for the AXIAL_AC Core Type" on page 26-532
"Defining Data for the AXIAL_PM Core Type" on page 26-548
"Defining the Shaft Data for a Generic Rotating Machine " on page 26-561
"Setting Up Analysis Parameters for a Generic Rotating Machine" on page 26-562
"Saving the Active RMxprt Project" on page 26-4
"Running an RMxprt Simulation " on page 26-142

Generic Rotating Machine Core Types

The Generic Rotating Machine (GRM) templates allow you to define any of several machine types using the core types listed below.

Core Type	Description
SLOT_AC	Slotted core with AC winding
SLOT_CAGE	Slotted core with squirrel-cage damper
SALNT_POLE	Salient-pole core with field winding
PM_INTERIOR	Interior permanent magnet core
SOLID	Solid core with/without copper/hysteresis sleeve
AXIAL_AC	Axial-flux slotted core with AC winding
AXIAL_CAGE	Axial-flux slotted core with squirrel-cage damper
AXIAL_PM	Axial-flux permanent magnet core
NONS_RELU	Non-salient reluctance core without winding

Note	Refer to "Defining a Generic Rotating Machine" on page 26-504 for detailed information on selecting GRM templates and defining the corresponding machine types that use these core types.

Related Topics

"Defining a Generic Rotating Machine" on page 26-504

Defining Data for the SLOT_AC Core Type

The general procedure for defining a SLOT_AC core type is as follows:

1. Double-click the **Machine>Stator** (or **Machine>Rotor**, if applicable) entry in the project tree to define data for the stator (or rotor). The **Stator** (or **Rotor**) **Properties**dialog box contains the following fields:

Number of Poles	The number of poles on which the winding is wound. Default is 2.
Number of Slots	The number of slots of the iron core. Default is 18.
Circuit Type	The drive circuit type. Select from the following six types (default is Y3): • Y3: Y-Type, 3-Phase • L3: Loop-Type, 3-Phase • S3: Star-Type, 3-Phase • C2: Cross-Type, 2-Phase • L4: Loop-Type, 4-Phase • S4: Star-Type, 4-Phase **Note** When you place the mouse cursor over the circuit type, a circuit diagram of the selected type appears in the dialog,
Slot Type	The slot type of the iron core. Click the **Slot Type** button to open the **Select Slot Type** dialog box. Select the desired type. Default is Type 1. **Note** When you place the mouse cursor over the slot type, a diagram of the selected type appears in the dialog box, displaying the slot dimension variables. Optionally, check **User Defined Slot** if you wish to define the slot dimensions using the **Slot Editor**.
Position Control	Determines if the drive circuit is controlled by position signals from a position sensor. (Checked for **IPM Brushless DC Motor** core type. Unchecked for all other types.)

2. Double-click the **Core** entry in the project tree to define data for the stator core. The core data **Properties** dialog box contains the following fields:

Outer Diameter	The outer diameter of the core.
Inner Diameter	The inner diameter of the core.
Length	The length of the core.
Stacking Factor	The effective magnetic length of the core. Default is 0.95.
Steel Type	The steel type of the core. Click the button to open the **Select Definition** window and select the desired material.
Press Board Thickness	The thickness of the pole press boards.
Magnetic Press Board	Check the box if the press board is made of magnetic material.
Skew Width	The skew width measured in degrees.
Lamination Sectors	The number of lamination sectors.

3. Double-click the **Slot** entry in the project tree to define the stator slot dimensions. Enter the available slot dimensions.

   An image of the selected slot type showing the dimension labels is displayed on the Diagram tab. The dimensions properties that appear depend on the **Slot Type** (selected above) and on whether or not **Auto Design** or **Parallel Tooth** is selected:

Auto Design	Check the box to have RMxprt auto design the Hs2, Bs1, and Bs2 dimensions.
Parallel Tooth	Check the box to have RMxprt design Bs1 and Bs2 based on **Tooth Width**. – Available only when **Auto Design** is cleared.
Tooth Width	The tooth width for parallel tooth. – Available only when **Parallel Tooth** is selected.
Hs0	– Always available.
Hs1	– Available only when the slot type is 2, 3, 4, 5, or 6.
Hs2	– Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.
Bs0	– Available only when the slot type is 1, 2, 3, 4, or 5.
Bs1	– Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.

Bs2	– Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Rs	– Available only when the slot type is 3 or 4.

4. Double-click the **Winding** entry in the project tree to define the conductors, windings, and insulation data for the winding
5. For an IPM Brushless DC Motor stator only, double-click the **Circuit** entry in the project tree to define the properties for the position control circuit.
6. Optionally, you can right-click on Core in the project tree to insert or remove Vent data.

Defining Data for the SLOT_CAGE Core Type

The general procedure for defining a SLOT_CAGE core type is as follows:

1. Double-click the **Machine>Rotor** entry in the project tree to define data for the rotor. In the **Rotor Properties**dialog box, enter the number of slots in the **Number of Slots** field. There are no additional settings.
2. Double-click the **Core** entry in the project tree to define data for the rotor core. The core data **Properties** dialog box contains the following fields:

Outer Diameter	The outer diameter of the core.
Inner Diameter	The inner diameter of the core.
Length	The length of the core.
Stacking Factor	The effective magnetic length of the core. Default is 0.95.
Steel Type	The steel type of the core. Click the button to open the **Select Definition** window and select the desired material.
Press Board Thickness	The thickness of the pole press boards.
Magnetic Press Board	Check the box if the press board is made of magnetic material.
Skew Width	The skew width measured in degrees.
Half Slot	Check the box to set the rotor damper slots to half slots.
Double Cage	Check the box to add a bottom damper.

3. Double-click the **Machine>Rotor>Core>Damper** (or **Bottom Damper**) entry in the project tree to open the **Properties** dialog box. (You can also enter values for the following properties in the **Properties** window of the desktop without opening a separate dialog box.)

Slot Type	The slot type of the damper. Click the **Slot Type** button

	to open the **Select Slot Type** dialog box. Select the desired type. Default is Type 1. **Note** When you place the mouse cursor over the slot type, a diagram of the selected type appears in the dialog box, displaying the slot dimension variables.
Cast Cage	Check the box if the cage is cast. – Not available for **Bottom Damper**.
Bar Fitting Gap	The gap between the slot edge and bar edge.
Bar End Extension	The single-side end extended bar length. – Not available for **Bottom Damper**.
End Ring Width	The axial length of the end ring. – Not available for **Bottom Damper**.
End Ring Height	The radial height of the end ring. – Not available for **Bottom Damper**.
Bar Conductor Type	Click the button to open the **Select Definition** window and select the desired conductor material.
End Ring Conductor Type	Click the button to open the **Select Definition** window and select the desired conductor material. – Not available for **Bottom Damper**.

4. Double-click the **Damper>Slot** (and **Bottom Damper>Slot**, if used) entry in the project tree to define the slot dimensions. Enter the slot dimensions.

   An image of the selected slot type showing the dimension labels is displayed on the Diagram tab. Below are the slot dimensions properties that appear:

Hs0	Slot dimension Hs0.
Hs01	Slot dimension Hs01.
Hs2	Slot dimension Hs2.
Bs0	Slot dimension Bs0.
Bs1	Slot dimension Bs1.
Bs2	Slot dimension Bs2.

5. Optionally, you can right-click on Core in the project tree to insert or remove Vent data.

Defining Data for the SOLID Core Type

The general procedure for defining a SOLID core type is as follows:

1. Double-click the **Machine>Rotor>Core**) entry in the project tree to define data for the core. The **Core Properties**dialog box contains the following fields:

Outer Diameter	The outer diameter of the core.

Inner Diameter	The inner diameter of the core.
Length	The length of the core.
Stacking Factor	The effective magnetic length of the core. Default is 0.95.
Steel Type	The steel type of the core. Click the button to open the **Select Definition** window and select the desired material.
Sleeve Type	Select the Sleeve type: • **No Sleeve** No Sleeve is selected by default if the machine type is **Solid-Rotor Induction**. • **Hysteresis** Hysteresis will be selected by default if the machine type is **Hysteresis Machine**. • **Conductor**
Sleeve Thickness	The sleeve thickness. Available only when the **Sleeve Type** is either **Hysteresis** or **Conductor**.
Sleeve Material	The sleeve material type of the core. Click the button to open the **Select Definition** window and select the desired material. If the **Sleeve Type** is **Hysteresis**, the message manager informs you if your choice is not a hysteresis type material. Similarly, If the **Sleeve Type** is **Conductor**, the message manager informs you if your choice is not a conductor type material. (Available only when the **Sleeve Type** is either **Hysteresis** or **Conductor**.)

Defining Data for the AXIAL_AC Core Type

The general procedure for defining an AXIAL_AC core type is as follows:

1. Double-click the **Machine>Stator** entry in the project tree to define data for the stator. The **Stator Properties**dialog box contains the following fields:

Number of Poles	The number of poles on which the winding is wound. Default is 2.
Number of Slots	The number of slots of the iron core. Default is 18.
Circuit Type	The drive circuit type. Select from the following six types (default is Y3):

	• Y3: Y-Type, 3-Phase • L3: Loop-Type, 3-Phase • S3: Star-Type, 3-Phase • C2: Cross-Type, 2-Phase • L4: Loop-Type, 4-Phase • S4: Star-Type, 4-Phase **Note** When you place the mouse cursor over the circuit type, a circuit diagram of the selected type appears in the dialog,
Slot Type	The slot type of the iron core. Click the **Slot Type** button to open the **Select Slot Type** dialog box. Select the desired type. Default is Type 1. **Note** When you place the mouse cursor over the slot type, a diagram of the selected type appears in the dialog box, displaying the slot dimension variables. Optionally, check **User Defined Slot** if you wish to define the slot dimensions using the **Slot Editor**.
Position Control	Determines if the drive circuit is controlled by position signals from a position sensor. (Checked for **Axial-Flux Brushless DC Motor** core type. Unchecked for all other types.)

2. Double-click the **Core** entry in the project tree to define data for the stator core. The core data **Properties** dialog box contains the following fields:

Outer Diameter	The outer diameter of the core.
Inner Diameter	The inner diameter of the core.
Length	The length of the core.
Stacking Factor	The effective magnetic length of the core. Default is 0.95.
Steel Type	The steel type of the core. Click the button to open the **Select Definition** window and select the desired material.

3. Double-click the **Slot** entry in the project tree to define the stator slot dimensions. Enter the available slot dimensions.

   An image of the selected slot type showing the dimension labels is displayed on the Diagram tab. The dimensions properties that appear depend on the **Slot Type** (selected above) and on whether or not **Auto Design** or **Parallel Tooth** is selected:

Auto Design	Check the box to have RMxprt auto design the Hs2, Bs1, and Bs2 dimensions.

Parallel Tooth	Check the box to have RMxprt design Bs1 and Bs2 based on **Tooth Width**. – Available only when **Auto Design** is cleared.
Tooth Width	The tooth width for parallel tooth. – Available when **Parallel Tooth** is selected.
Hs0	– Always available.
Hs1	– Available only when the slot type is 2, 3, 4, 5, or 6.
Hs2	– Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically.
Bs0	– Available only when the slot type is 1, 2, 3, 4, or 5.
Bs1	– Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Bs2	– Available only when **Auto Design** is cleared. When **Auto Design** is selected, this slot dimension is determined automatically. When **Parallel Tooth** is selected, this slot dimension is determined based on the value entered in the **Tooth Width** field.
Rs	– Available only when the slot type is 3 or 4.

4. Double-click the **Winding** entry in the project tree to define the conductors, windings, and insulation data for the winding
5. For an Axial-Flux Brushless DC Motor stator only, double-click the **Circuit** entry in the project tree to define the properties for the position control circuit.

Defining Data for the AXIAL_CAGE Core Type

The general procedure for defining a AXIAL_CAGE core type is as follows:

1. Double-click the **Machine>Rotor** entry in the project tree to define data for the rotor. In the **Rotor Properties**dialog box, enter the number of slots in the **Number of Slots** field. There are no additional settings.
2. Double-click the **Core** entry in the project tree to define data for the rotor core. The core data **Properties** dialog box contains the following fields:

Outer Diameter	The outer diameter of the core.
Inner Diameter	The inner diameter of the core.
Length	The length of the core.
Stacking Factor	The effective magnetic length of the core. Default is 0.95.

Steel Type	The steel type of the core. Click the button to open the **Select Definition** window and select the desired material.
Press Board Thickness	The thickness of the pole press boards.
Magnetic Press Board	Check the box if the press board is made of magnetic material.
Skew Width	The skew width measured in degrees.
Half Slot	Check the box to set the rotor damper slots to half slots.
Double Cage	Check the box to add a bottom damper.

3. Double-click the **Machine>Rotor>Core>Damper** (or **Bottom Damper**) entry in the project tree to open the **Properties** dialog box. (You can also enter values for the following properties in the **Properties** window of the desktop without opening a separate dialog box.)

Slot Type	The slot type of the damper. Click the **Slot Type** button to open the **Select Slot Type** dialog box. Select the desired type. Default is Type 1. **Note** When you place the mouse cursor over the slot type, a diagram of the selected type appears in the dialog box, displaying the slot dimension variables.
Cast Cage	Check the box if the cage is cast. – Not available for **Bottom Damper**.
Bar Fitting Gap	The gap between the slot edge and bar edge.
Bar End Extension	The single-side end extended bar length. – Not available for **Bottom Damper**.
End Ring Width	The axial length of the end ring. – Not available for **Bottom Damper**.
End Ring Height	The radial height of the end ring. – Not available for **Bottom Damper**.
Bar Conductor Type	Click the button to open the **Select Definition** window and select the desired conductor material.
End Ring Conductor Type	Click the button to open the **Select Definition** window and select the desired conductor material. – Not available for **Bottom Damper**.

4. Double-click the **Damper>Slot** (and **Bottom Damper>Slot**, if used) entry in the project tree to define the slot dimensions. Enter the slot dimensions.

   An image of the selected slot type showing the dimension labels is displayed on the Diagram tab. Below are the slot dimensions properties that appear:

Hs0	Slot dimension Hs0.
Hs01	Slot dimension Hs01.
Hs2	Slot dimension Hs2.
Bs0	Slot dimension Bs0.
Bs1	Slot dimension Bs1.
Bs2	Slot dimension Bs2.

Defining Data for the SALNT_POLE Core Type

The general procedure for defining a SALNT_POLE core type is as follows:

1. Double-click the **Machine>Rotor** (or **Machine>Stator**, if applicable) entry in the project tree to define data for the rotor (or stator). In the **Rotor** (or **Stator**) **Properties**dialog box, enter the number of poles in the **Number of Poles** field (default is 2). There are no additional settings.
2. Double-click the **Core** entry in the project tree to define data for the rotor or stator core. The core data **Properties** dialog box contains the following fields:

Outer Diameter	The outer diameter of the core.
Inner Diameter	The inner diameter of the core.
Length	The length of the core.
Stacking Factor	The effective magnetic length of the core. Default is 0.95.
Steel Type	The steel type of the core. Click the button to open the **Select Definition** window and select the desired material.
Press Board Thickness	The thickness of the pole press boards.
Magnetic Press Board	Check the box if the press board is made of magnetic material.
Skew Width	The skew width measured in degrees.
Pole Type	Click the **Pole Type** button to open the **Select Pole Type** dialog box. The SALNT_POLE rotor core type supports one pole type: Type 1; while the SALNT_POLE stator supports two pole types. A diagram displaying the pole dimension variables appears on the Diagram tab. **Note** When you place the mouse cursor over the pole type, a diagram of the selected type appears in the dialog box, displaying the pole dimension variables.

3. Double-click the **Machine>Rotor>Core>Pole** or **Machine>Stator>Core>Pole**entry in the project tree to open its **Properties** dialog.

Second Air Gap	The second air gap between the pole body and the yoke.
Pole Body Width	Pole body width dimension.
Pole Body Height	Pole body height dimension.
Pole Shoe Width	Pole shoe width dimension.
Pole Shoe Height	Pole shoe height dimension.
Pole Arc Offset	The pole-arc center offset from the rotor center.
Second Pole Arc	Check if you want the pole surface to include a second arc or line in a half-pole range. When checked, enables Off2_x and Off2_y.
Off2_x	Offset of the second arc perpendicular to the pole-center line. – Available when **Second Pole Arc** is checked.
Off2_y	Offset of the second arc in parallel with the pole-center line. – Available when **Second Pole Arc** is checked.
Stacking Factor	The effective magnetic length of the rotor pole.
Steel Type	Steel material type. Click the button to open the **Select Definition** window and select the steel material type from the list. Use the **Material Filter** tab settings to filter for **Steel** materials.
Bp1	Pole dimension Bp1. – Available only for stator core **Pole Type** 1.
Bp2	Pole dimension Bp2. – Available only for stator core **Pole Type** 1.
Rp1	Pole dimension Rp1. – Available only for stator core **Pole Type** 2.
Rp2	Pole dimension Rp2. – Available only for stator core **Pole Type** 2.

4. Optionally, you can add a damper by right-clicking **Core** in the project tree and selecting **Insert Damper**. You can then double-click the **Machine>Rotor>Core>Damper** entry in the project tree to open the **Properties** dialog box. (You can also enter values for the following properties in the **Properties** window of the desktop without opening a separate dialog box.)

Damper Slots Per Pole	The number of damper slots per pole.

Slot Type	The slot type of the damper. Click the **Slot Type** button to open the **Select Slot Type** dialog box. Select the desired type. Default is Type 1. **Note** When you place the mouse cursor over the slot type, a diagram of the selected type appears in the dialog box, displaying the slot dimension variables.
Cast Rotor	Check the box if the cage is cast.
Bar Conductor Type	Click the button to open the **Select Definition** window and select the desired conductor material.
End Length	The single-side end bar length. – Available only when the **End Ring Type** is 2 or 3.
End Ring Width	The axial length of the end ring. – Available only when the **End Ring Type** is 2 or 3.
End Ring Height	The radial height of the end ring. – Available only when the **End Ring Type** is 2 or 3.
End Ring Conductor Type	Click the button to open the **Select Definition** window and select the desired conductor material. – Not available for **Bottom Damper**.
End Ring Type	Click the button to open the **Select Definition** window and select the desired end ring type. **Note** When you place the mouse cursor over the end ring type, an explanation appears in the dialog box, describing the end ring.
Slot Pitch	The slot pitch for the damper slots.
Center Slot Pitch	The center slot pitch for the damper slots.
Bar Fitting Gap	The gap between the slot edge and bar edge.

5. If a damper is present, double-click the **Damper>Slot** entry in the project tree to define the slot dimensions. Enter the slot dimensions.

   An image of the selected slot type showing the dimension labels is displayed on the Diagram tab. Below are the slot dimensions properties that appear:

Hs0	Slot dimension Hs0.
Hs01	Slot dimension Hs01.
Hs2	Slot dimension Hs2.
Bs0	Slot dimension Bs0.
Bs1	Slot dimension Bs1.
Bs2	Slot dimension Bs2.

6. Double-click the **Machine>Rotor>Field** or **Machine>Stator>Field** entry in the project tree to open its **Properties** dialog and define the Field Data.
7. Double-click the **Machine>Rotor>Field>Winding** or **Machine>Stator>Field>Winding** entry in the project tree to open its **Properties** dialog and define the Field Winding Data.
8. Optionally for machine rotors only, you can right-click on **Machine>Rotor** in the project tree and select **Insert Brush** to define brush data. Double-click **Brush** in the project tree to open its **Properties** dialog box contains the following fields:

Diameter	Brush surface diameter.
Brush Width	Brush width.
Brush Length	Brush length.
Brush Drop	Voltage drop of a brush.
Contact Resistance	Contact resistance of a brush.
Brush Press	Brush pressure per unit area.
Frictional Coefficient	Frictional coefficient of a brush.

Defining SALNT_POLE Stator and Rotor Field Data for a Generic Rotating Machine.

To define the stator and rotor windings and insulation data:

1. To open the Stator or Rotor Field **Properties** window, double-click the **Machine-Stator-Field** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Enter the thickness of the insulation under the pole shoe in the **Shoe Insulation** field.
3. Enter the thickness of the insulation at the pole body side in the **Pole Insulation** field.
4. Enter the minimum gap in the **Winding Clearance** field. The winding clearance is one of the following: the minimum gap between two field windings, or the minimum gap between a field winding and a commutating winding.
5. Enter the exciter efficiency value in the **Exciter Efficiency** field.
6. Enter the exciting current value in the **Exciting Current** field.
7. Click **OK** to close the **Properties** window.

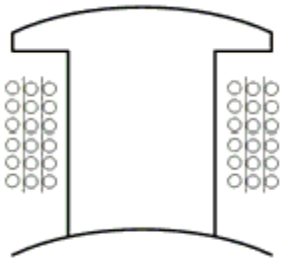
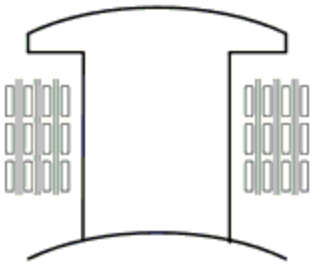
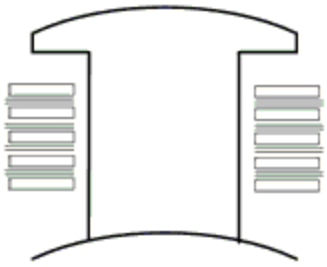
Defining SALNT_POLE Stator and Rotor Field Winding Data for a Generic Rotating Machine.

Note	Not applicable for SALNT_STEP stators.

1. To open the rotor or stator field winding **Properties** dialog box, double-click the **Machine>Stator>Field>Winding** or **Machine>Rotor>Field>Winding** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate dialog box.)

2. Select a **Winding Type**:
   a. Click the button for **Winding Type** to open the **Winding Type** window.
   b. Select **Round**, **Cylinder**, or **EdgeWise** winding.

   When you place the mouse cursor over a winding button, an outline of the selected winding appears. The following table describes the three types of windings:

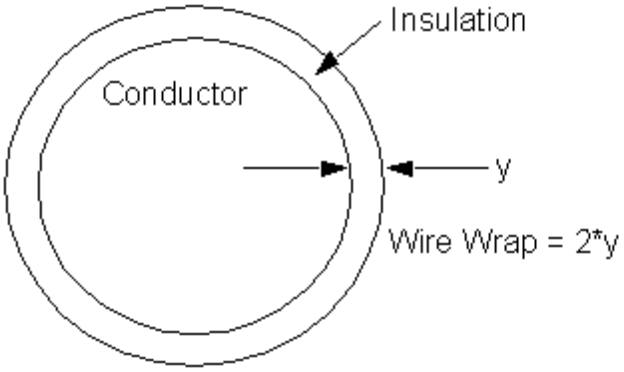
Type	Description
Round	A Round winding: 
Cylinder	A Cylinder winding: 
EdgeWise	An EdgeWise winding: 

3. Once you have selected a winding, click **OK** to close the **Winding Type** dialog box and return to the **Properties** window.
4. Enter the number of parallel branches in one phase of the winding in the **Parallel Branches** field.

5. Enter the total number of conductors for each pole in the **Conductors per Pole** field. Odd numbers for input and output leads on different sides. Enter 0 to have RMxprt auto-design this value.
6. For Round and Cylinder winding types, enter the number of wires per conductor in the **Number of Strands** field. Enter 0 to have RMxprt auto-design this value.

Note	The **Number of Strands** field is invisible (not used) when the EdgeWise winding type has been selected.

7. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



8. For an **EdgeWise** winding type, enter the thickness of the inter-turn insulation in the **Inter-turn Insulation** field.

Note	The **Inter-turn Insulation** field is invisible (not used) when either the Round or Cylinder winding type has been selected.

9. Select the **Wire Size**:
   a. Click the button for **Wire Size** to open the **Wire Size** dialog box.
   b. Select a value from the **Wire Diameter** pull-down list.
   c. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

<*number*>	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.
USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.
AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.

MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21mm and 2 with a diameter of 0.13mm.

The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>General Options>Machines** command.

d. When you are done setting the wire size, click **OK** to close the **Wire Size** dialog box and return to the **Properties** dialog box.

10. When you are done setting the wire size, click **OK** to close the **Wire Size** dialog box and return to the **Properties** dialog box.
11. Click the **Conductor Type** button to open the **Select Definition** window **Materials** tab and selecting the desired conductor material type. Use the **Material Filter** tab settings to filter for **Conductor** materials. Click **OK** to close the window.
12. Enter the axial gap between the field winding and pole body in the **Axial Clearance** field.
13. Enter the inner radius at winding fillet in the **Winding Fillet** field.
14. Enter the inner diameter of the coil tip in the **Tip Inner Diameter** field.
15. Enter the limited cross-section width for winding design or arrangement in the **Limited Cross Width** field. Enter 0 for maximum available area.
16. Enter the limited cross-section height for winding design or arrangement in the **Limited Cross Height** field. Enter 0 for maximum available area.
17. Click **OK** to close the **Properties** dialog box.

Defining Data for the NONS_RELU Core Type

The general procedure for defining a NONS_RELU core type is as follows:

1. Double-click the **Machine>Rotor** entry in the project tree to define data for the rotor. In the **Rotor Properties**dialog box, enter the number of poles in the **Number of Poles** field (default is 2). There are no additional settings.
2. Double-click the **Core** entry in the project tree to define data for the rotor or stator core. The core data **Properties** dialog box contains the following fields:

Outer Diameter	The outer diameter of the core.
Inner Diameter	The inner diameter of the core.
Length	The length of the core.
Stacking Factor	The effective magnetic length of the core. Default is 0.95.
Steel Type	The steel type of the core. Click the button to open the **Select Definition** window and select the desired material.

Press Board Thickness	The thickness of the pole press boards.
Magnetic Press Board	Check the box if the press board is made of magnetic material.
Skew Width	The skew width measured in degrees.
Pole Type	Click the **Pole Type** button to open the **Select Pole Type** dialog box. The NONS_RELU rotor core type supports four pole types: • 1 – Laminated • 2 – Concentric-polyline • 3 – Hyperbolic-curve • 4 – Hyperbolic-polyline Default type is 4 – Hyperbolic-polyline. **Note** When you place the mouse cursor over the pole type, a diagram of the selected type appears in the dialog.

3. Double-click the **Machine>Rotor>Core>Pole** entry in the project tree to open its **Properties** dialog box, which contains the following fields:

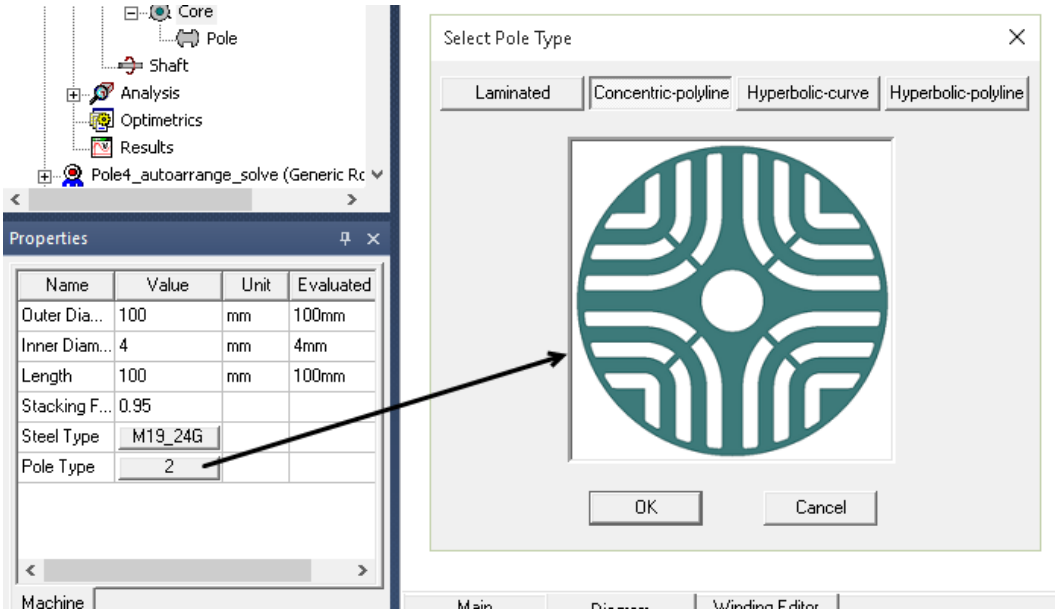
Note	The currently selected pole type diagram displays on the machine editor window Diagram tab.

Barriers per Pole	Number of barriers in one pole. Invisible when pole type is 1 (Laminated).
H	Bridge thickness. Invisible when pole type is 1 (Laminated).
W	Rib width at the barrier center. Invisible when pole type is 1 (Laminated).
R	Barrier fillet radius. Invisible when pole type is 1 (Laminated).
R0	Radius of the bottom barrier arch for Pole Type 2 (Concentric-polyline); or line connecting arch radius for Pole Type 4 (Hyperbolic-polyline). Visible only when Pole Type is 2 or 4.
Rb	In-circle radius of the bottom barriers.
B0	The first (from inner to outer) barrier width at the barrier center. Invisible when pole type is 1 (Laminated).
Y0	The first yoke width (between the first and the

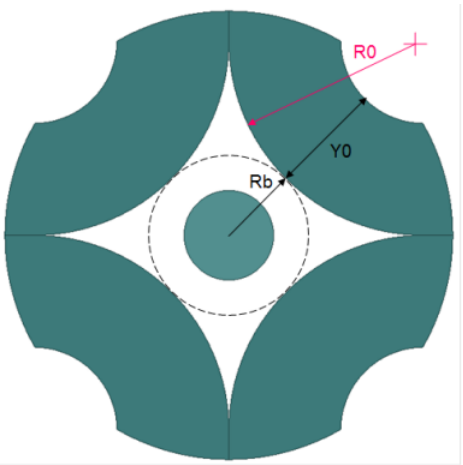
	second barriers) at the barrier center.
Barrier Auto Arrangement	Check the box to let barriers be auto-arranged. Invisible when pole type is 1 (Laminated).
Extra Barriers	For pole types 2, 3, or 4, if the number of **Barriers per Pole** is greater than 1, and **Barrier Auto Arrangement** has not been enabled, the **Extra BarriersSetup** button opens the **Barriers Setup** dialog in which you can specify the **Bi** (bottom widths of the additional barriers) and **Yi** (bottom widths of the additional yokes) dimensions. Invisible if the number of **Barriers per Pole** is 1, or if **Barrier Auto Arrangement** has been enabled.

NONS_RELU Rotor Core Pole Types

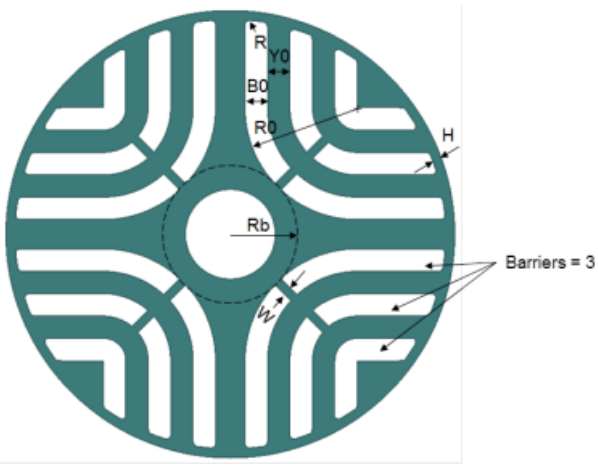
The NONS_RELU rotor core type supports four pole types. You can choose the pole type by selecting the **Machine>Rotor>Core** entry in the project tree, then clicking the **Pole Type** button in its Properties window to open the **Select Pole Type** window.



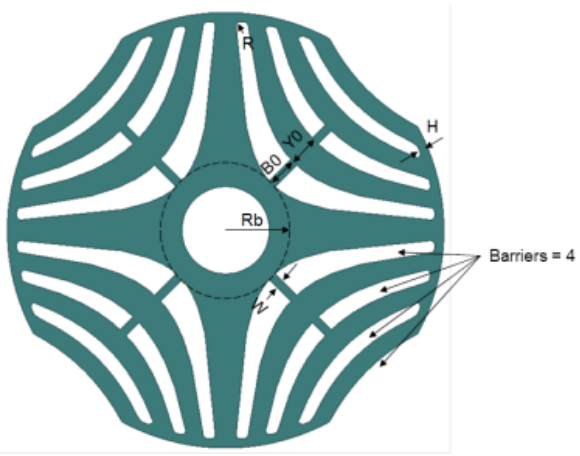
The four available pole types are shown below. Refer to *Defining Data for the NONS_RELU Core Type* for details on defining the various pole properties.



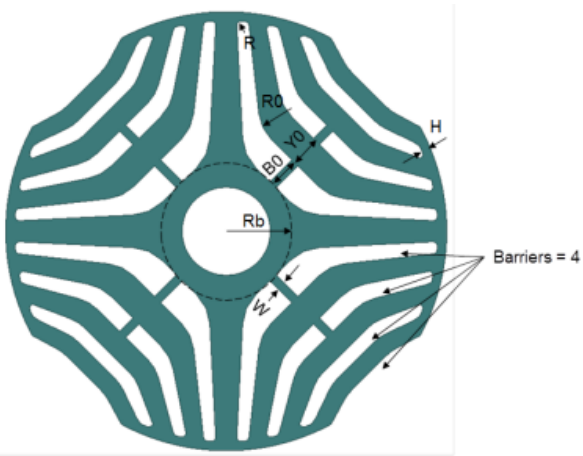
Laminated (Type 1)



Concentric-polyline (Type 2)



Hyperbolic-curve (Type 3)



Hyperbolic-polyline (Type 4)

Defining Data for the PM_INTERIOR Core Type

The general procedure for defining a PM_INTERIOR core type is as follows:

1. Double-click the **Machine>Rotor** entry in the project tree to define data for the rotor. In the **Rotor Properties**dialog box, enter the number of poles in the **Number of Poles** field (default is 4). There are no additional settings.
2. Double-click the **Core** entry in the project tree to define data for the rotor or stator core. The core data **Properties** dialog box contains the following fields:

Outer Diameter	The outer diameter of the core.
Inner Diameter	The inner diameter of the core.
Length	The length of the core.
Stacking Factor	The effective magnetic length of the core. Default is 0.95.
Steel Type	The steel type of the core. Click the button to open the **Select Definition** window and select the desired material.
Pole Type	Click the **Pole Type** button to open the **Select Pole Type** dialog box. The PM_INTERIOR rotor core type supports six pole types: Default type is 3. **Note** When you place the mouse cursor over the pole type, a diagram of the selected type appears in the dialog.

3. Double-click the **Machine>Rotor>Core>Pole** entry in the project tree to open its **Properties** dialog box, which contains the following fields:

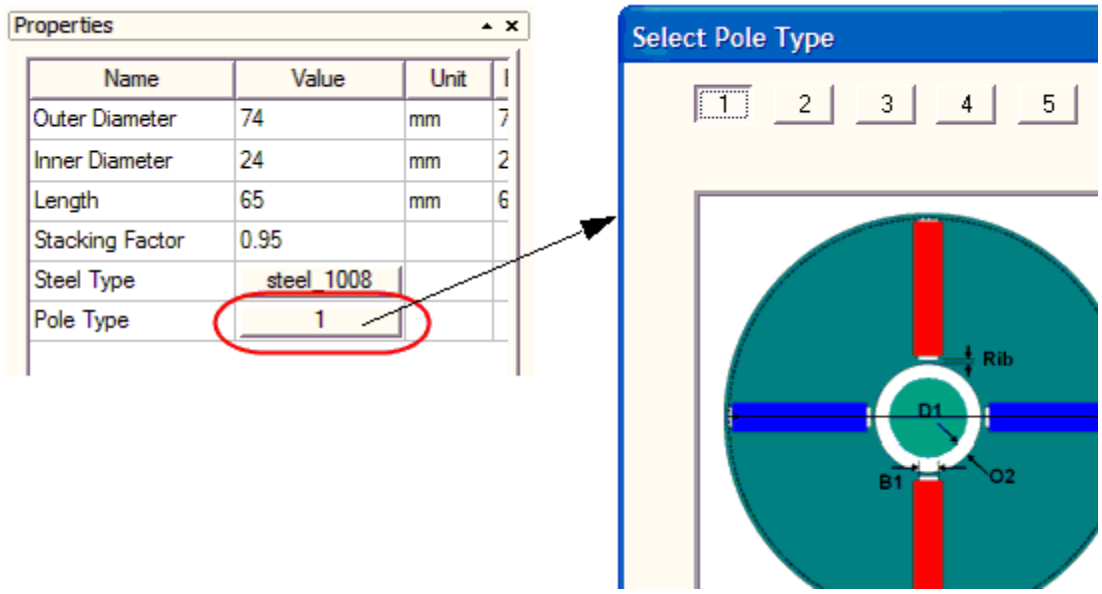
Note	The currently selected pole type diagram displays on the machine editor window Diagram tab.

D1	Limited diameter for magnet ducts.
O1	Magnet duct dimension. Invisible when pole type is 1 or 2.
O2	Magnet duct dimension.
B1	Magnet duct dimension.
Rib	Magnet duct dimension.
HRib	Magnet duct dimension. Invisible when pole type is 1,2, or 6.
Layers	Number of duct layers.
Layer Pitch	Pitch between two duct layers.
Magnet Thickness	Magnet thickness, or duct thickness.
Magnet Width	Total width of all magnets per pole.
Magnet Type	Magnet material type. Click the button to open the **Select Definition** window and select the magnet material type from the list. Use the **Material Filter** tab settings to

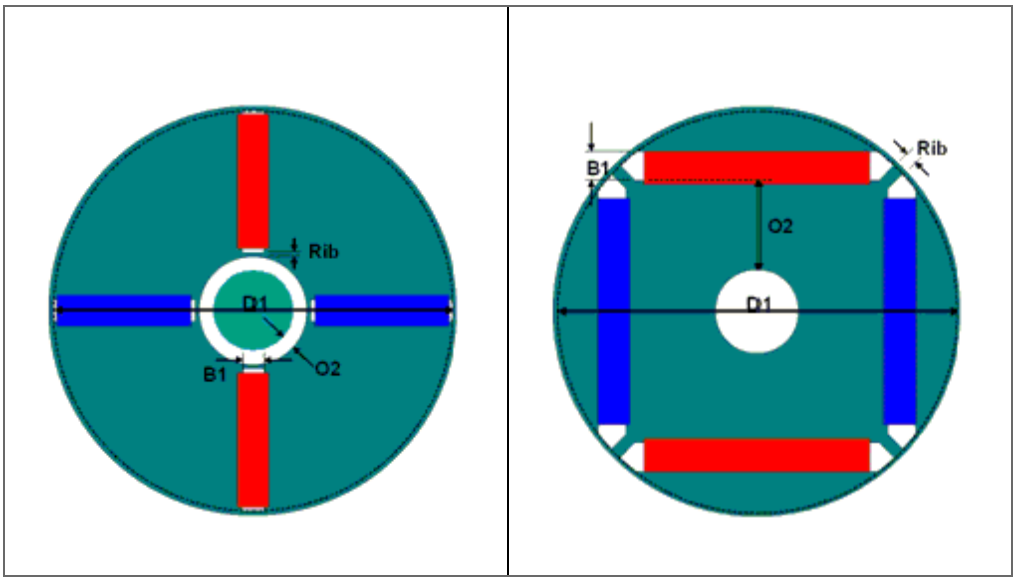
	filter for **Magnet** materials.

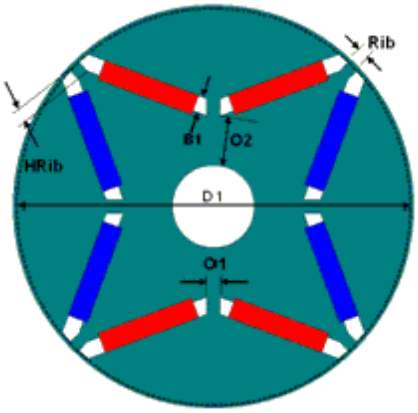
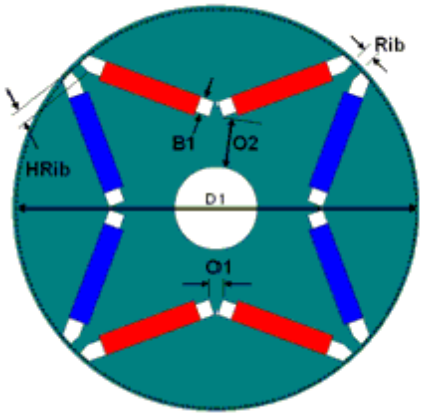
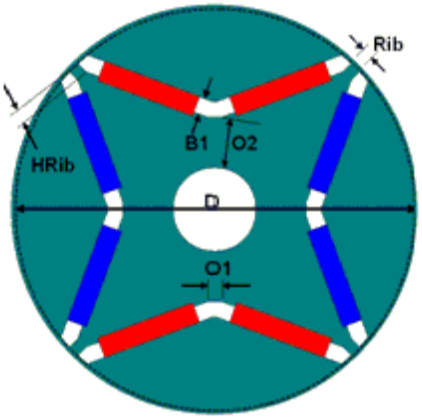
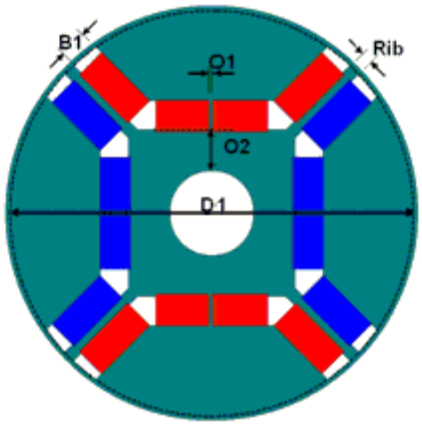
PM_INTERIOR Rotor Core Pole Types

The PM_INTERIOR rotor core type supports six pole types. You can choose the pole type by selecting the **Machine>Rotor>Core** entry in the project tree, then clicking the **Pole Type** button in its Properties window to open the **Select Pole Type** window.



The six available pole types are shown below. Refer to *Defining Data for the PM_INTERIOR Core Type* for details on defining the various pole properties.



Pole Type 1	Pole Type 2
	
Pole Type 3	**Pole Type 4**
	
Pole Type 5	**Pole Type 6**

Defining Data for the AXIAL_PM Core Type

The general procedure for defining a AXIAL_PM core type is as follows:

1. Double-click the **Machine>Rotor** entry in the project tree to define data for the rotor. In the **Rotor Properties**dialog box, enter the number of poles in the **Number of Poles** field (default is 2). There are no additional settings.

2. Double-click the **Core** entry in the project tree to define data for the rotor or stator core. The core data **Properties** dialog box contains the following fields:

Outer Diameter	The outer diameter of the core.
Inner Diameter	The inner diameter of the core.
Length	The length of the core.
Stacking Factor	The effective magnetic length of the core. Default is 0.95.
Steel Type	The steel type of the core. Click the button to open the **Select Definition** window and select the desired material.
Skew Width	The skew width measured in degrees.

3. Double-click the **Machine>Rotor>Core>Pole** entry in the project tree to open its **Properties** dialog box, which contains the following fields:

Embrace	Pole embrace value.
Magnet Type	Magnet material type. Click the button to open the **Select Definition** window and select the magnet material type from the list. Use the **Material Filter** tab settings to filter for **Magnet** materials.
Magnet Length	The radial length of the magnet.
Magnet Thickness	Axial thickness of the magnet, per side.

Defining the Stator and Rotor Windings for a Generic Rotating Machine

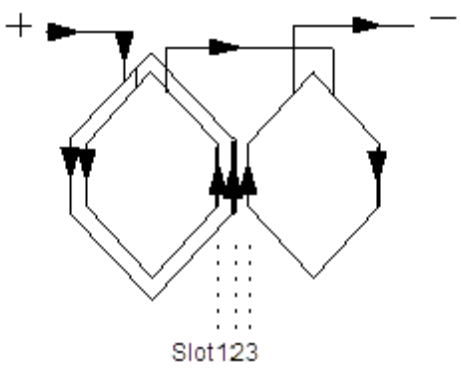
Note	This section is not applicable to: • AXIAL_PM type stators or rotors. • PM_INTERIOR, AXIAL_CAGE, and SOLID type rotors. • SALNT_POLE type stators and rotors. Refer to *Defining SALNT_POLE Stator and Rotor Field Winding Data for a Generic Rotating Machine* for winding information for SALNT_POLE stators and rotors. • SALNT_STEP type stators.

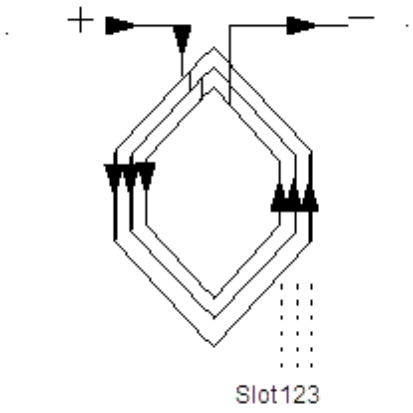
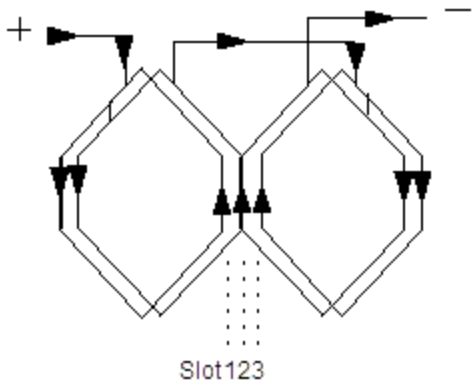
To define the wires, conductors, insulation, and windings of a stator or rotor:

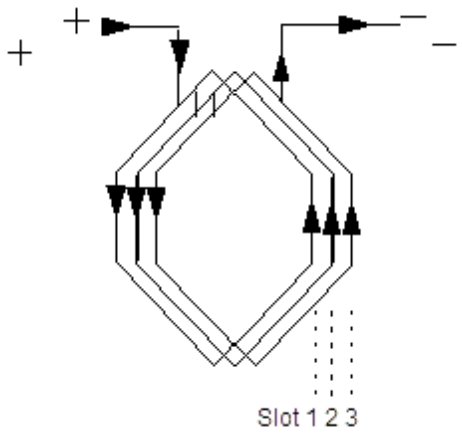
1. To open the rotor or stator slot winding **Properties** dialog box, double-click the **Machine>Stator>Winding** or **Machine>Rotor>Winding** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate dialog box.)
2. Click the **Winding** tab.

3. Choose the desired number of layers in the winding from the drop-down list in the **Winding Layers** field.
4. Select a **Winding Type**:
   a. Click the button for **Winding Type**.
      The **Winding Type** window appears.
   b. Select from one of the following three types of winding:
      - **Whole Coiled**
      - **Half Coiled**
      - **Editor**

When you place the mouse cursor over a winding button, an outline of the selected winding appears. The following table describes the six types of windings that are possible (three for one-layer and three for two-layer):

Type	Description
Winding Editor (one-layer)	A user-defined one-layer winding arrangement. You need to set up the winding arrangement for each slot in the Winding Editor.
Whole-Coiled (one-layer)	A one-layer whole-coiled winding: 
Half-Coiled (one-layer)	A one-layer concentric half-coiled winding:

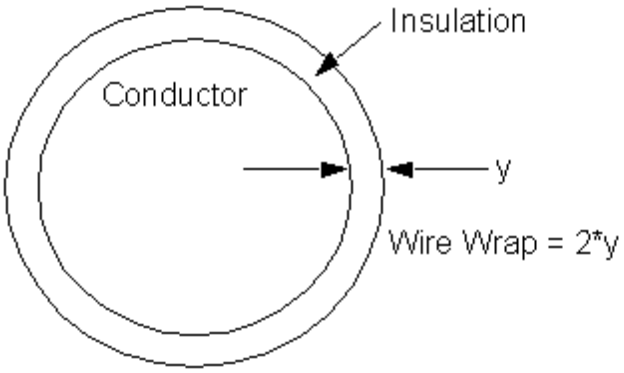
	
Winding Editor (two-layer)	A user-defined two-layer winding arrangement. When you select for winding layers you can specify a different winding arrangement for each slot in the Winding Editor.
Whole-Coiled (two-layer)	A two-layer whole-coiled winding:  The phase belt for this winding configuration is equal to 360/2*m*, where *m* is the phase number.
Half-Coiled (two-layer)	A two-layer half-coiled winding:

	 There is only one coil per phase per pair of poles.

Note	For a two-layer winding, if you check Constant Pitch in the Winding Editor, only the top layer needs to be defined; the bottom layer is determined according to the coil pitch.

Once you have clicked a button to select a winding, click **OK** to close the **Winding Type** dialog box and return to the **Properties** window.

5. Enter the number of parallel branches in one phase of the winding in the **Parallel Branches** field.
6. Enter the total number of conductors in each slot in the **Conductors per Slot** field. This value is the number of turns per coil multiplied by the number of layers. Enter 0 to have RMxprt auto-design this value.
7. Enter the coil pitch, measured in number of slots, in the **Coil Pitch** field. The coil pitch is the number of slots separating one winding. For example, if a coil starts in slot 1 and ends in slot 6, it has a coil pitch of 5. This field is not displayed when the number of **Winding Layers** is 1.
8. Enter the number of wires per conductor in the **Number of Strands** field. Enter 0 to have RMxprt auto-design this value.
9. Enter the thickness of the double-sided wire wrap in the **Wire Wrap** field. Enter 0 to automatically obtain this value from the wire library.



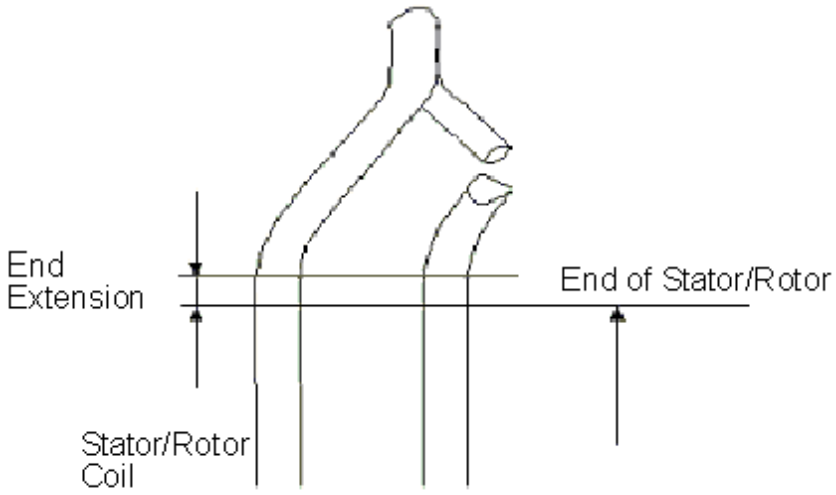
10. Select the **Wire Size**:
    a. Click the button for **Wire Size**.
       The **Wire Size** dialog box appears.
    b. Select a value from the **Wire Diameter** pull-down list.
    c. Select a wire gauge from the **Gauge** pull-down menu. You can select from the following options:

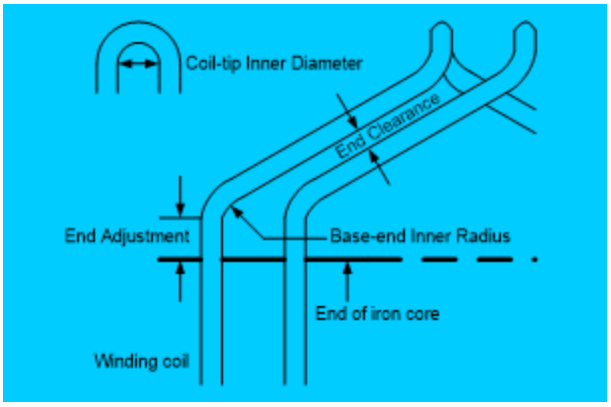
< ***number*** **>**	You can select a specific gauge number. When you select a gauge number, the **Wire Diameter** field is automatically updated.
USER	This option allows you to manually enter the **Wire Diameter**. This is useful when you want to enter a diameter that does not correspond to a particular wire gauge.
AUTO	This option sets the **Wire Diameter** to zero, and RMxprt automatically calculates the optimal value. The diameter information is then written to the output file when you analyze the design.
MIXED	This option allows you to define a conductor that is made of different size wires. For example, a single conductor may consist of 5 wires, 3 wires with a diameter of 0.21mm and 2 with a diameter of 0.13mm.

       The gauge number is based on AWG settings. You can create your own wire table using **Machine>Wire**, and then you can select this wire table using the **Tools>Options>Machine Options** command.

    When you are done setting the wire size, click **OK** to close the **Wire Size** dialog box and return to the **Properties** dialog box.
11. Click the button for **Conductor Type**. This opens the **Select Definition** dialog box in which you select the material type to be used for the winding.
12. Click the **End/Insulation** tab. Select or clear the **Input Half-turn Length** check box. Do one of the following:

- If you selected **Input Half-turn Length**, then enter the half-turn length of the armature winding in the Half Turn Length field.
- If you cleared **Input Half-turn Length**, then enter the end length adjustment of the stator/rotor coils in the **End Extension** field. The end extension is the distance one end of the conductor extends vertically beyond the end of the stator or rotor.





13. Enter the span length correction factor to scale the end span length in the **Correction Factor** field.
14. Enter the inner radius of the base corner in the **Base Inner Radius** field.
15. Enter the inner diameter of the coil tip in the **Tip Inner Diameter** field.
16. Enter the distance between two adjacent coils in the **End Clearance** field.
17. Enter the thickness of the single-side coil wrap insulation in the **Coil Wrap** field.
    This field is applicable and displayed only for slot types 5 and 6.
18. Enter the thickness of the slot liner insulation in the **Slot Liner** field.
19. Enter the thickness of the wedge insulation in the **Wedge Thickness** field.

20. Enter the thickness of the insulation layer in the **Layer Insulation** field.
    This field is applicable and displayed only when the **Winding Layers** value is 2.
21. Enter the bottom insulation thickness in the **Bottom Insulation** field.
    This field is applicable and displayed only for slot types 5 and 6.
22. Enter the limited slot fill factor for the wire design in the **Limited Fill Factor** field. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.
    This field is applicable and displayed only for slot types 1, 2, 3, and 4.
23. Enter the top spare slot space for a dual-winding machine in the **Top Spare Space** field.
24. Enter the bottom spare slot space for a dual-winding machine in the **Bottom Spare Space** field.
25. Click **OK** to close the **Properties** dialog box.

Stator and Rotor Winding Data for Generic Rotating Machines

Note	This section is not applicable to: • AXIAL_PM type stators or rotors. • PM_INTERIOR, AXIAL_CAGE, and SOLID type rotors. • SALNT_POLE type stators and rotors. Refer to *Defining SALNT_POLE Stator and Rotor Field Winding Data for a Generic Rotating Machine* for winding information for SALNT_POLE stators and rotors.

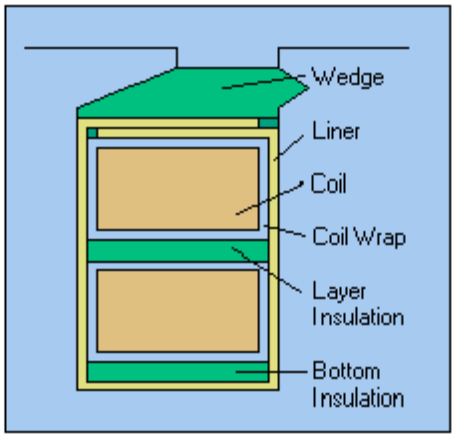
To access the core slot data, double-click either the **Machine>Rotor>Core>Slot** or the **Machine>Rotor>Core>Slot** entry in the project tree.

The winding data **Properties** dialog box contains the following fields:

Winding tab	**Winding Layers**	The number of winding layers. (The bottom layer is for another side slot if the core is double-sided.) Select 1 or 2 in the pull-down list. Default value is 2. **Winding Layers** is always 2 if the core is double-sided.
	Winding Type	The type of stator winding. Click the button to open the **Winding Type** window and choose from **Whole-Coiled**, **Half-Coiled**, and **Editor**. Default value is **Whole-Coiled**.
	Parallel Branches	The number of parallel branches in the winding. Default value is 1.
	Conductors per Slot	The number of conductors per slot (0 for auto-design).
	Coil Pitch	The coil pitch measured in number of slots. This field is displayed only when the number of **Winding Layers** is 2. **Coil Pitch** is always 0 if the core is double-sided.

	Number of Strands	The number of wires per conductor (0 for auto-design). Default value is 1.
	Wire Wrap	The thickness of the double-sided wire wrap (0 to automatically obtain this value from the wire library).
	Wire Size	The diameter of the wire (0 for auto-design). Click the button to open the **Wire Size** dialog box where you can specify units, wire type, diameter, and gauge.
End/Insulation tab	**Input Half-turn Length**	Select or clear this check box to specify whether or not you want to enter the half-turn length. When this check box is selected, the **Half Turn Length** field appears the next time you open the **Properties** window. When this check box is selected, the **End Extension** field appears instead. Default value is unchecked.
	Half Turn Length	The average half-turn length of the armature winding. Visible only when **Input Half Turn Length** is checked.
	End Extension	The end length adjustment of the coils, which is the distance one end of the conductor extends vertically beyond the end of the stator or rotor. Invisible when **Input Half Turn Length** is checked.
	Correction Factor	End span length correction factor to scale the end span length. Must be > 0. Default value is 1.0. Invisible when **Input Half Turn Length** is checked
	Base Inner Radius	The inner radius of the base corner.
	Tip Inner Diameter	The inner diameter of the coil tip. Invisible if the core is double-sided.
	End Clearance	The end clearance between two adjacent coils.
	Coil Wrap	The thickness of the single-side coil wrap. This field is displayed only when the **Slot Type** is 5 or 6.
	Slot Liner	The thickness of the slot liner insulation.
	Wedge Thickness	The thickness of the wedge insulation.
	Layer Insulation	The thickness of the insulation layer. Invisible when number of winding layers is 1 or the core is double-sided.
	Bottom Insulation	Thickness of the bottom insulation. This field is displayed only when the **Slot Type** is 5 or 6.
	Limited Fill Factor	The limited slot fill factor for the wire design. The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area. This field is displayed only for Slot Types 1, 2, 3, or 4.

	Top Spare Space	The top spare slot space for a dual-winding machine. The value must be greater-than or equal-to 0 and less-than 1.
	Bottom Spare Space	The bottom spare slot space for a dual-winding machine. The value must be greater-than or equal-to 0. Also, the sum of the **Top Spare Space** and **Bottom Spare Space** values must less-than 1.
	Conductor Type	Conductor material type of the Stator/Rotor Winding.



Winding Editor for a Generic Rotating Machine

For a generic rotating machine, you may want to specify a different number of conductors for each stator or rotor slot. The **Winding Editor** enables you to specify the number of turns for each coil. To enable the **Winding Editor**, you must have set the **Winding Property** for the **Winding Type** to **Editor**.

To specify the number of turns for each coil:

1. Click **Machine>Winding>Edit Layout**.

   The **Winding Editor** dialog box appears.

2. In the table in the upper left, set which **Phase** you want for each coil and which slot is the “In” and “Out” slot for the current in each coil.
3. If you are working on a quarter or half model, you may want to specify a multiplier by selecting a value from the **Periodic Multiplier** drop-down menu.

4. Select or deselect the **Constant Turns** or **Constant Pitch** check boxes, depending on whether you want to be able to change these setting in the table above. When these options are selected, you cannot change the turns or pitch.
5. When you are satisfied with the coil settings, click **OK** to close the **Winding Editor** dialog box.

Defining Different Size Wires for a Generic Rotating Machine

Use the **Gauge** option if you have a conductor that is made up different size wires.

To define different size wires:

1. In the **Wire Size** window, select **MIXED** from the **Gauge** pull-down menu.
2. Select either **Round** or **Rectangular** as the **Wire Type**.
3. Enter the appropriate wire data in the table:
   - For a round wire:
     - Click **Add** to add the new wire data.
     - Enter the **Diameter** in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this diameter.
   - For a rectangular wire:
     - Click **Add** to add the new wire data.
     - Enter the **Width** of the wire in the table. The width should be greater than the thickness.
     - Enter the **Thickness** of the wire in the table.
     - Enter the **Fillet** value in the table.
     - Enter a **Number** in the table to specify how many of the conductor's wires have this data.
4. Repeat step 3 for each size wire you want to add.
5. When you are finished defining the wires, click **OK** to close the **Wire Size** window.

Note	For example, if one conductor is made up of 5 wires, and 3 of those wires have a diameter of 0.21mm, and the other 2 have a diameter of 0.13mm, then the mixed wire size table will have two lines. The first line will list Diameter = 0.21 and Number = 3. The second line will list Diameter = 0.13 and Number = 2. An equivalent wire diameter is displayed as Wire Size value in the Winding tab in the Properties window.

Defining the Stator and Rotor Circuits for a Generic Rotating Machine

You can define stator and rotor Circuit parameters for generic rotating machines whose Source Type is DC.

Note	This procedure is not applicable to: • AXIAL_PM type stators or rotors. • SALNT_POLE type stators or rotors.

	• AXIAL_AC, AXIAL_CAGE, SLOT_CAGE, and SOLID type rotors. • PM_INTERIOR type rotors.

To define the circuit parameters of a stator or rotor:

1. To define the circuit parameters click either the stator or rotor Circuit icon in the project tree. You can then edit the brush parameters directly in the Properties window. (You can also open the rotor or stator circuit **Properties** dialog box by double-clicking the **Machine>Stator>Circuit** or **Machine>Rotor>Circuit** entry in the project tree on the desktop.)
2. Select the **Control Type** from the drop down menu. Choices are: DC, CCC, PWM, and HCC.
3. If the stator or rotor **Position Control** has been enabled, set the **Lead Angle of Trigger** value.
4. If the **Control Type** chosen is either DC or CCC, set the **Trigger Pulse Width**. The default value is 120.
5. Set the **Transistor Drop** (voltage drop of one transistor).
6. Set the **Diode Drop**.
7. If the **Control Type** is CCC, set the **Maximum Current** and **Minimum Current** values.
8. If the **Control Type** is PWM, set the **Modulation Index** and **Carrier Frequency Times**.
9. If the **Control Type** is HCC, set the **Reference Amplitude** and **Hysteresis Band**.

Stator and Rotor Circuit Data for Generic Rotating Machines

To access the stator or rotor circuit data, double-click either the **Machine>Stator>Circuit** or the **Machine>Rotor>Circuit** entry in the project tree.

Note	This section is not applicable to PM_INTERIOR and SALNT_STEP rotors .

The circuit data **Properties** dialog box contains the following fields:

Control Type	Control Type: DC, CCC (chopping current control), PWM, HCC (hysteresis current control). Default is DC. **Note** For SALNT_STEP Stators, only DC and CCC control types are available.
Lead Angle of Trigger	Lead angle of trigger in electrical degrees. Visible only when the machine stator or rotor **Position Control** is enabled.
Trigger Pulse Width	Trigger pulse width in electrical degrees. Visible only when **Control Type** is DC or CCC. Default value is 120.
Transistor Drop	Voltage drop of one transistor.

Diode Drop	Voltage drop of one diode, or the total voltage for star-type circuits in the discharge loop.
Maximum Current	Maximum current for chopping current control. Visible only when **Control Type** is CCC.
Minimum Current	Minimum current for chopping current control. Visible only when **Control Type** is CCC.
Modulation Index	Modulation index (the ratio of the sine-wave amplitude to the triangular amplitude). Visible only when **Control Type** is PWM.
Carrier Frequency Times	Carrier frequency times (the ratio of the triangular frequency to the sine-wave frequency). Visible only when **Control Type** is PWM.
Reference Amplitude	The amplitude of the sine-wave reference current. Visible only when **Control Type** is HCC.
Hysteresis Band	The difference between the upper and lower hysteresis limits. Visible only when **Control Type** is HCC.

Defining the Axial AC Rotor Brush for a Generic Rotating Machine

Optionally, you can insert or remove brush data for generic rotating machines that have an Axial AC Rotor Structure. If you have inserted a brush, the icon appears under the core slot in the project tree.

To insert a brush:

1. Right-click on the rotor core icon to display the pop-up menu.
2. Click **Insert Brush**.
3. To edit the brush data, double-click the brush icon to open the brush properties dialog.

   The brush data **Properties** dialog box contains the following fields:

Diameter	Brush surface diameter.
Brush Width	Brush width.
Brush Length	Brush length.
Brush Drop	Voltage drop of a brush.
Contact Resistance	Contact resistance of a brush.
Brush Press	Brush pressure per unit area.
Frictional Coefficient	Frictional coefficient of a brush.

To remove an existing brush:

1. Right click on the rotor icon to display the pop-up menu.
2. Click **Remove Brush**.

Vent Data for Generic Rotating Machines

Optionally, you can insert or remove Vent data for generic rotating machines that have either an Inner Rotor or Outer Rotor Structure. If you have inserted a Vent, the icon appears under the core slot in the project tree.

Note	• This section does not apply to the SALNT_POLE, NONS_RELU, or SOLID rotor types.

To insert a vent:

1. Right-click on the stator or rotor core icon to display the pop-up menu.
2. Click **Insert Vent**.

To remove an existing vent:

1. Right click on the stator or rotor icon to display the pop-up menu.
2. Click **Remove Vent**.

The vent data **Properties** dialog box contains the following fields.

Vent Ducts	Number of radial vent ducts. Default is 0.
Duct Width	Width of radial vent ducts.
Magnetic Spacer Width	Width of magnetic spacer which hold vent ducts. 0 for non-magnetic spacer.
Duct Pitch	Vent ducts
Holes per Row	Number of axial vent holes per row Not available for SLOT_CAGE rotor type.
Inner Hole Diameter	Diameter of vent holes in inner row. Not available for SLOT_CAGE rotor type.
Outer Hole Diameter	Diameter of vent holes in outer row. Not available for SLOT_CAGE rotor type.
Inner Hole Location	Center-to-center diameter of inner row hole vents. Not available for SLOT_CAGE rotor type.
Outer Hole Location	Center-to-center diameter of outer row hole vents. Not available for SLOT_CAGE rotor type.

Defining the Shaft Data for a Generic Rotating Machine

To define the shaft:

1. To open the shaft data **Properties** dialog box, double-click the **Machine>Shaft** entry in the project tree on the desktop. (You can also enter values in the **Properties** section of the desktop without opening a separate window.)
2. Select or clear the **Magnetic Shaft** check box to specify whether or not the shaft is to be made of magnetic material.
3. Enter the frictional loss in the **Frictional Loss** field.
4. Enter the windage loss (or power for wind power generators) measured at the **Reference Speed** in the **Windage Loss or Power** field.
5. Enter the reference speed at which frictional and windage losses are measured in the **Reference Speed** field.
6. Click **OK** to close the **Properties** window.

Shaft Data for General DC Machines

To access the shaft data, double-click the **Machine>Shaft** entry in the project tree.

The **Shaft Data Properties** window contains the following fields:

Magnetic Shaft	Select or clear this check box to indicate whether or not the shaft is made of magnetic material. When selected, the shaft is magnetic. Default is unchecked for PM_INTERIOR rotor type.
Frictional Loss	The frictional loss measured at the **Reference Speed**.
Windage Loss or Power	The **Windage Loss** (or **Power** for wind power generators) measured at the **Reference Speed**.
Reference Speed	The speed at which the friction and windage losses are measured. Default is 3600 rpm for PM_INTERIOR and AXIAL_CAGE rotor types.

Setting Up Analysis Parameters for a Generic Rotating Machine

To define solution parameters for a generic rotating machine:

1. Right-click **Analysis** in the project tree, and click **Add Solution Setup**.

   The **Solution Setup** dialog box appears.
2. Click the **General** tab.
   a. If you wish to change the automatically assigned setup name, enter a name for the setup in the **Setup Name** field.
   b. The solution setup is enabled by default. Uncheck the **Enabled** box to disable the setup, if desired.
   c. Select an operation type (**Motor**, **Generator**, or **Wind Generator**) from the **Operation Type** pull-down list. The **Operation Type** is set to **Motor** by default.
   d. Select the **Load Type** used in the machine.

- If the **Operation Type** is **Motor**, select one of the following **Load Type** options:

Const Speed	The speed remains constant in the motor.
Const Power	The output power remains constant in the motor. (Default type.)
Const Torque	The torque remains constant regardless of the speed. In this case, $T_{load} = T_{rated}$, given by the output power divided by the given rated speed.
Linear Torque	The torque increases linearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})$ where T_{rated} is given by the output power divided by the given rated speed.
Fan Load	The load varies nonlinearly with speed. In this case, $T_{load} = T_{rated} * (n/n_{rated})^2$ where T_{rated} is given by the output power divided by the given rated speed.

- If the **Operation Type** is either **Generator** or **Wind Generator**, select one of the following **Load Type** options:

Infinite Bus	When the generator is connected to an infinite bus of the power system, the terminal voltages remain constant and the output power is determined by the output currents.
Independent Generator	The generator supplies electrical power to a load inductor. For an AC generator, the resistance and inductance of the load inductor are determined by the rated output power, the rated voltage, and the rated power factor. For a DC generator, the resistance of the load resistor is determined by the rated output power and the rated voltage.

e. Enter the rated mechanical or electrical output (apparent) power in the **Rated Output Power** field.

When the **Operation Type** is **Generator** and **Source Type** is **AC**, the property name changes to **Rated Apparent Power (kVA)**, and the unit column is blank.

f. Enter the applied or output rated voltage in the **Rated Voltage** field. (RMS value for AC source.) Default value is 208 V.

g. Enter the temperature at which the system functions in the **Operating Temperature** field. Default is 75 cel.

3. Click the **Generic Rotating Machine (GRM)** tab.

a. For AC source type machines, enter the rated power factor in the **Rated Power Factor** field. The default value is 0.8. This field is invisible when the **Source Type** is **DC**.

b. For AC source type machines, if you wish to determine load impedance when the phase current leads the phase voltage, enable the **Capacitive Power Factor** check box. This field is invisible when the **Source Type** is **DC**.

c. Enter the source frequency in the **Frequency** field and select the appropriate unit of measure. The default value is 60 Hz. This field is invisible when the **Source Type** is **DC**.

d. Optionally, you can click the **Use Defaults** button to restore the tab settings to default values.

4. Optionally, click the **Defaults** tab. This tab contains two buttons:
   - Click the **Save Defaults** button to save the currently-defined settings as defaults for future setups.
   - Click the **Revert to Standard Defaults** button to clear existing user-defined defaults and revert to the standard settings.
5. Click **OK** to close the **Solution Setup** window.

Related Topics

Solution Data for Generic Rotating Machines

Solution Data for Generic Rotating Machines

To access the solution data, double-click the solution setup located under **Analysis** in the project tree to open the solution **Properties** dialog box. Solution data is also accessible in the desktop **Properties** window for the selected setup.

The solution setup Properties contains the following fields:

General tab:

Name	The name of the setup. Not editable.
Enabled	Check box that enables/disables the solution setup. Default is checked (enabled)
Operation Type	Set the type of operation for the machine analysis. Pull-down list selections are: Motor, Generator, Wind Generator. Default is Motor.
Load Type	Select the mechanical or electrical load type from the pull-down list. For Motor **Operation Type** the selections are: Const Speed, Const Power, Const Torque, Linear Torque, Fan Load. Default is Const Power. For Generator and Wind Generator **Operation Type** the selections are: Infinite Bus and Independent Generator. Default is Infinite Bus.
Rated Output Power	Enter the rated mechanical or electrical output (apparent) power, and select the unit.
Rated Voltage	Enter a value for the rated voltage and select the unit.
Rated Speed	Type a value for the rated speed, and select the unit.
Operating Temperature	Type a value for the operating temperature, and select the unit.

GRM tab:

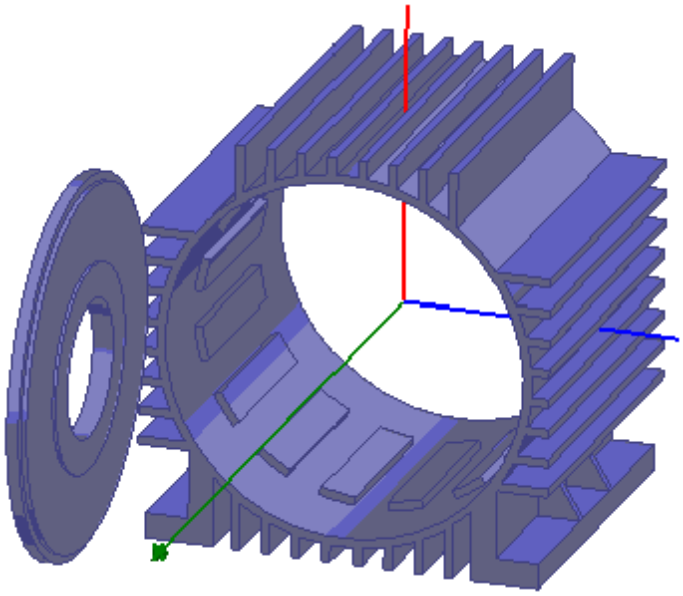
Rated Power Factor	Enter the rated power factor for AC Source Type. This field is displayed only for AC Source Type. Default value is 0.8.
Capacitive Power Factor	Check box that enables/disables use of a capacitive power factor. Used together with the **Rated Power Factor** when the phase current leads the phase voltage, to determine load impedance. Default is unchecked.
Frequency	Enter a value for the frequency, and select the unit.

Related Topics

Setting Up Analysis Parameters for a Generic Rotating Machine

Adding Machine Housings

You can add machine housing parts to all machine types. Please note that machine housings for Generic Rotating Machines and Single-Phase Induction Motors are available only for Inner Rotor type machines. RMxprt designs that include machine housings can be exported to **Motor-CAD™** for advanced thermal analysis.



1. To include a Housing in the design, right-click the Machine icon in the Project Manager tree and select **Insert Housing**. A new Housing icon is added to the machine design containing Front Cap and Rear Cap housing elements.
2. Right-click on Housing in the Project Manager tree to include and specify parameter data for the following housing elements:

- Duct
- Fin
- Flange
- Foot

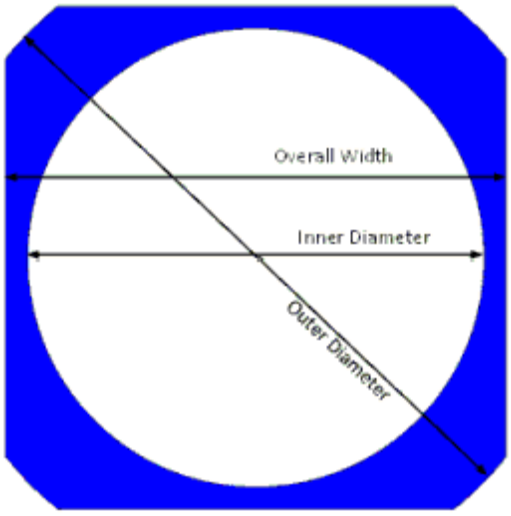
3. Specify properties for the Housing and related elements by selecting them and setting parameter data values either in the Properties window or Properties dialog.
4. You can remove the Housing and all related elements by right-clicking Machine and selecting **Remove Housing**. Individual housing elements can be removed by right-clicking Housing and selecting the desired **Remove** *<objectname>* menu item.

Note	The Front Cap and Rear Cap elements are always included with the Housing and cannot be removed individually.

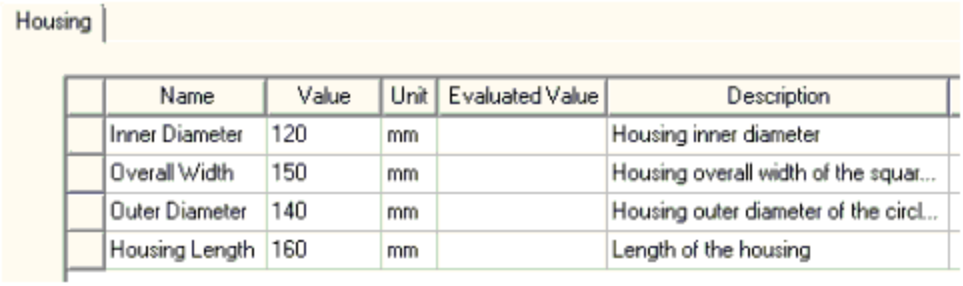
Related Topics

Housing Data
Housing Duct Data
Housing Fin Data
Housing Flange Data
Housing Foot Data
Housing Front Cap and Rear Cap Data
Exporting a MotorCAD Project

Housing Data



You can specify the following Housing properties:


Housing

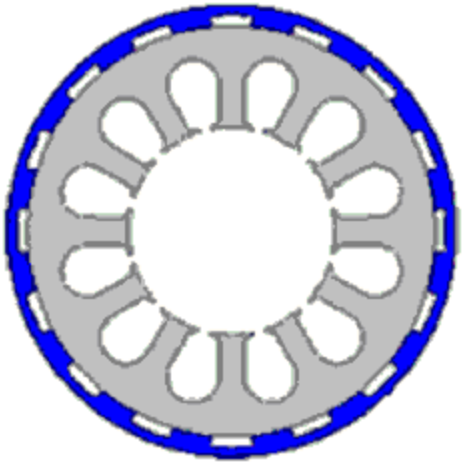
	Name	Value	Unit	Evaluated Value	Description
	Inner Diameter	120	mm		Housing inner diameter
	Overall Width	150	mm		Housing overall width of the squar...
	Outer Diameter	140	mm		Housing outer diameter of the circl...
	Housing Length	160	mm		Length of the housing

Name	Description	Notes
Inner Diameter	Housing inner diameter	Inner Diameter > 0
Outer Diameter	Housing outer diameter of the circle profile	Outer Diameter > Inner Diameter
Overall Width	Housing overall width of the square profile	Overall Width > Inner Diameter
Housing Length	Length of the housing	Housing Length > 0

Related Topics

Adding Machine Housings
Housing Duct Data
Housing Fin Data
Housing Flange Data
Housing Foot Data
Housing Front Cap and Rear Cap Data

Housing Duct Data



You can specify the following housing Duct properties:

Duct

	Name	Value	Unit	Evaluated Value	Description
	Number of Ducts	16			Number of ducts, or water jack
	Duct Type	Axial Wind Duct			Duct type
	Duct Height	5	mm		Duct height in radial direction
	Duct Embrace	0.5			Ratio of duct span, or duct widt...
	Duct Length	65	mm		Axial length for axial wind duct o...

Name	Description	Notes
Number of Ducts	Number of ducts, or water jackets	Number of Ducts > 0
Duct Type	Duct type	Available types: • 1-Axial Wind Duct • 2-Axial Water Jacket • 3-Spiral Water Jacket Default: Axial Wind Duct.
Duct Height	Duct height in radial direction	0 < Duct Height < Housing Thickness (half of the difference between the housing and stator outer diameters)
Duct	Ratio of duct span, or duct	0 < Duct Embrace < 1

Embrace	width, to duct pitch with uniform distribution	
Duct Length	Axial length for axial wind duct only	0 < Duct Length < Housing Length


Axial Wind Duct

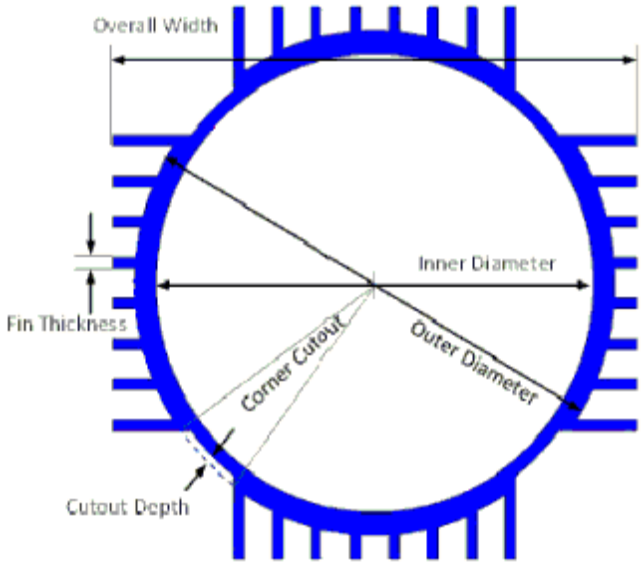

Axial Water Jacket


Spiral Water Jacket

Related Topics

Adding Machine Housings
Housing Data
Housing Fin Data
Housing Flange Data
Housing Foot Data
Housing Front Cap and Rear Cap Data

Housing Fin Data



You can specify the following housing Fin properties:

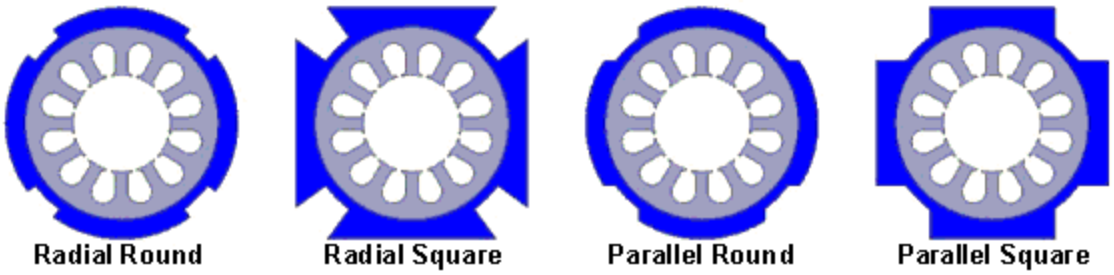
Fin

	Name	Value	Unit	Evaluated Value	Description
	Number of Fins	8			Number of fins per side (one fourth of ...
	Fin Type	Square Ring			Fin type
	Corner Cutout(%)	0.2			Corner cutout percentage
	Cutout Depth	1	mm		Corner cutout depth from Outer Diame...
	Fin Thickness	5	mm		Fin Thickness

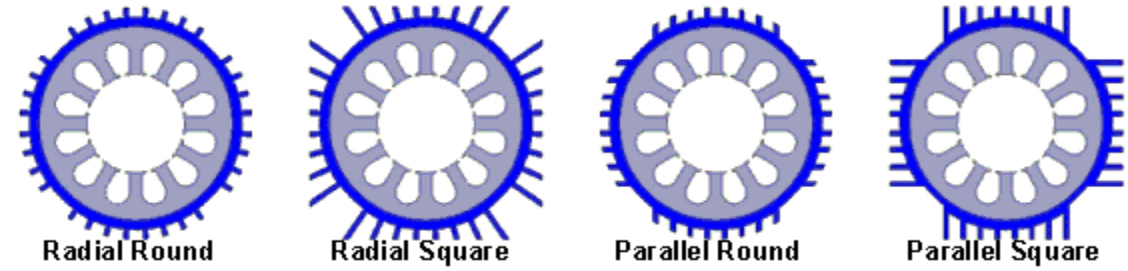
Name	Description	Notes
Number of Fins	Number of fins per side (one fourth of total fins)	Number of Fins > 0
Fin Type	Fin type	Available types: • 1-Radial Round • 2-Radial Square • 3-Parallel Round • 4-Parallel Square • 5-Square Ring • 6-Servo Ring.

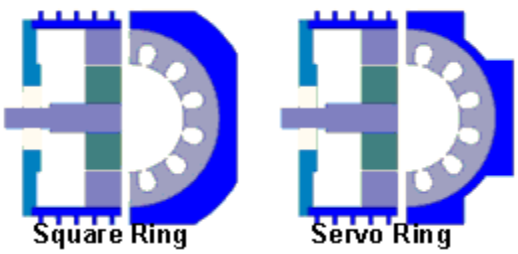
		Default: Radial Round (Types 5 & 6 are disabled if Number of Fins = 1).
Corner Cutout (%)	Corner cutout percentage	0 ≤ Corner Cutout < 1
Cutout Depth	Corner cutout depth from Outer Diameter, or Fin Extension for Fin Type 5	Cutout Depth ³ 0
Fin Thickness	Fin Thickness	Fin Thickness > 0

Fin Types when Number of fins = 1:



Fin types when number of fins = 2 or greater:

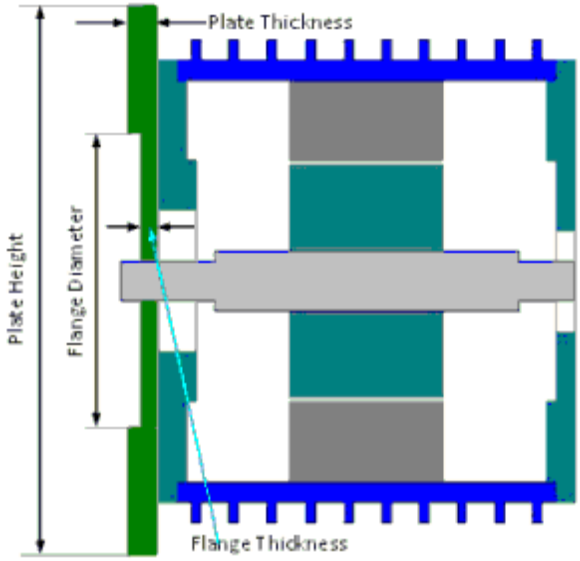




Related Topics

Adding Machine Housings
Housing Data
Housing Duct Data
Housing Flange Data
Housing Foot Data
Housing Front Cap and Rear Cap Data

Housing Flange Data



You can specify the following housing Flange properties:

Flange

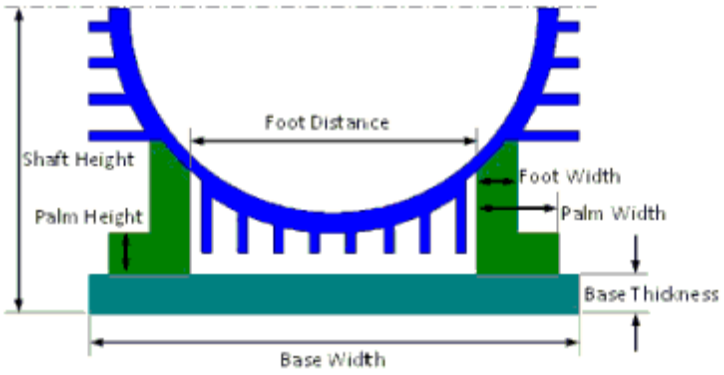
	Name	Value	Unit	Evaluated Value	Description
	Flange Diameter	100	mm		Diameter of the flange section that cover...
	Flange Thickn...	10	mm		Thickness of the flange section that cov...
	Plate Height	180	mm		Height of the cooling plate which the flan...
	Plate Thickness	20	mm		Thickness of cooling plate
	Plate Width	180	mm		Width of cooling plate

Name	Description	Notes
Flange Diameter	Diameter of the flange section that covers the front bearing	Flange Diameter > 0
Flange Thickness	Thickness of the flange section that covers the front bearing	Flange Thickness > 0
Plate Height	Height of the cooling plate which the flange mounts on	Plate Height > Flange Diameter
Plate Thickness	Thickness of cooling plate	Plate Thickness > Flange Thickness
Plate Width	Width of cooling plate	Plate Width > 0

Related Topics

Adding Machine Housings
Housing Data
Housing Duct Data
Housing Fin Data
Housing Foot Data
Housing Front Cap and Rear Cap Data

Housing Foot Data



You can specify the following housing Foot properties:

Foot

	Name	Value	Unit	Evaluated Value	Description
	Base Length	160	mm		Base plate length in axial direction
	Base Width	150	mm		Base plate width
	Base Thickness	20	mm		Base plate thickness
	Shaft Height	80	mm		Height of shaft from foot
	Foot Length	160	mm		Foot length in axial direction
	Foot Width	10	mm		Foot width at junction with housing
	Foot Distance	90	mm		Distance between two feet
	Palm Height	10	mm		Palm height
	Palm Width	30	mm		Palm width at junction with base plate

Name	Description	Notes
Base Length	Base plate length in axial direction	Base Length > 0
Base Width	Base plate width	Base Width > 0
Base Thickness	Base plate thickness	Base Thickness > 0
Shaft Height	Height of shaft from foot	Shaft Height > Base Thickness
Foot Length	Foot length in axial direction	0 < Foot Length < Base Length
Foot Width	Foot width at junction with housing	Foot Width > 0
Foot Distance	Distance between two feet	0 < Foot Distance < Base Width
Palm Height	Palm height	0 < Palm Height < (Shaft Height - Base Thickness)
Palm Width	Palm width at junction with base plate	Palm Width ≤ (Base Width - Foot Distance)/2 Palm Width ³ Foot Width

Related Topics

Adding Machine Housings

Housing Data

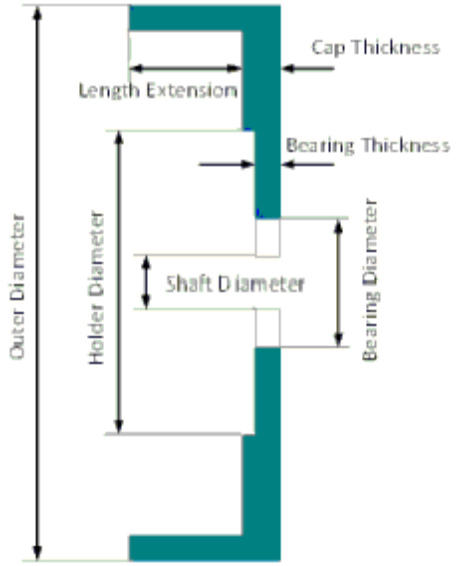
Housing Duct Data

Housing Fin Data

Housing Flange Data

Housing Front Cap and Rear Cap Data

Housing Front Cap and Rear Cap Data



You can specify the following housing Front Cap and Rear Cap properties:

	Name	Value	Unit	Evaluated Value	Description
	Outer Diameter	140	mm		Outer diameter of end cap
	Length Extension	-3	mm		Length extended from housing end to cap...
	Cap Thickness	10	mm		Axial thickness of end cap
	Holder Diameter	80	mm		Outer diameter of bearing holder
	Bearing Diameter	40	mm		Bearing outer diameter
	Bearing Thickness	15	mm		Axial thickness of bearing (or bearing hold...
	Shaft Diameter	22	mm		Shaft diameter at end cap
	Shaft Extension	0	mm		Shaft extension length from end cap to ou...

Name	Description	Notes
Outer Diameter	Outer diameter of end cap	Cap Outer Diameter ³ Housing Outer Diameter
Length Extension	Length extended from housing end to cap inner surface, < 0 for rabbet length	
Cap	Axial thickness of end cap	Cap Thickness > 0

Thickness		
Holder Diameter	Outer diameter of bearing holder	Holder Diameter > 0
Bearing Diameter	Bearing outer diameter	Bearing Diameter < Holder Diameter
Bearing Thickness	Axial thickness of bearing (or bearing holder)	Bearing Thickness > 0
Shaft Diameter	Shaft diameter at end cap	Shaft Diameter < Bearing Diameter
Shaft Extension	Shaft extension length from end cap to outside for torque output or fan installation	Shaft Extension ³ 0

Related Topics

Adding Machine Housings
Housing Data
Housing Duct Data
Housing Fin Data
Housing Flange Data
Housing Foot Data

Stator Vent Data

Select a Machine Type to get more information of Stator Vents:

- Three-Phase Induction Motors
- Three-Phase Synchronous Machines
- Three-Phase Non-Salient Synchronous Machines

Rotor Vent Data

Select a Machine Type to get more information of Rotor Vents:

- Three-Phase Induction Motors
- Three-Phase Non-Salient Synchronous Machines

27 - Using the Maxwell Circuit Editor

To open Maxwell Circuit Editor:

- Click **Start>Programs>Ansys Electromagnetics>Ansys Electromagnetics Suite [version]>Ansys Maxwell Circuit Editor**.

The following menus are available in Maxwell Circuit Editor:

File menu
Edit menu
View menu
Projectmenu
Draw menu
Schematic menu
Maxwell Circuit menu
Toolsmenu
Windowmenu
Help menu

Related Topics:

Schematic Editor

The Component Libraries Window

Copying and Pasting Properties

Maxwell Circuit Editor Component Models

Placing Components in the Maxwell Circuit Editor Schematic

Assigning Component Properties in Maxwell Circuit Editor

Schematic Editor

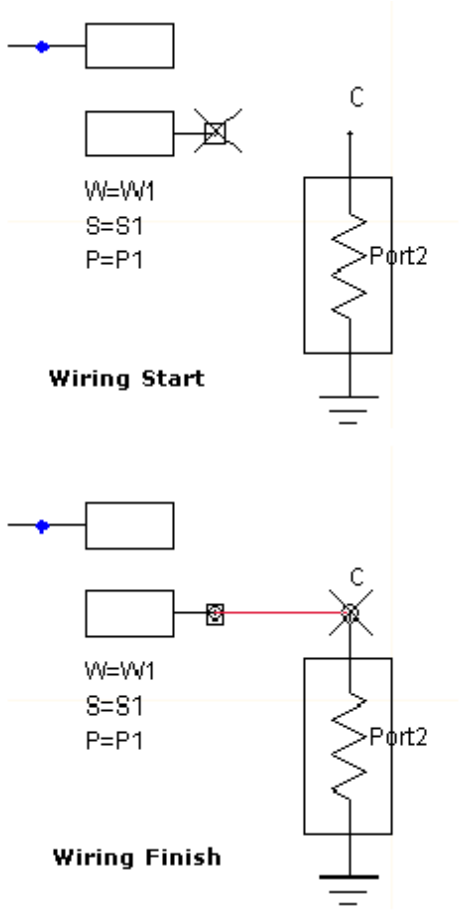
The Schematic editor is the tool for creating circuit schematics, or designs for a transient solution type. A design graphically represents and captures the electrical structure and characteristics of a circuit. You create such a design by starting the schematic editor and placing components, ports, connectors, and wires into a default empty schematic.

The Schematic Editor Window

The **Schematic Editor** window allows you to place components and wire them together. You can move components by simply selecting and dragging them. **Copy** and **Paste** can be used on

components and their wires within the schematic editor. You can also copy and paste to other schematics.

As you place the cursor near a pin of a component, it changes from an arrow to an **X**. This indicates that the schematic editor is in the wiring mode. In the wiring mode, click to start drawing a wire. Click again to end the wire.



Commonly used items such as ports, grounds, and page connectors can be placed in the schematic by selecting them from the **Draw** menu.

Controls to **Zoom In**, **Zoom Out**, and **Fit Drawing** to the editor window are available on the **View** menu, the **View** ribbon tab, and on the shortcut menu that opens when you right-click in a schematic.

If your mouse has a wheel, you can zoom while moving the mouse wheel up (zoom in) or down (zoom out). You can scroll vertically by holding down the **Shift** key while moving the mouse wheel or scroll horizontally by holding down the **Ctrl** key while moving the mouse wheel. This behavior can be changed using the **Zoom with mouse wheel** setting located in the Options dialog box, Schematic Editor/General section. If **Zoom with mouse wheel** is set to false (unchecked), you can scroll vertically with the mouse wheel, and zoom with the mouse wheel by pressing the **Shift** key.

The **arrow keys** scroll the view up, down, left, or right in small increments. The page up and page down keys scroll the view up or down in larger increments. If your mouse has a scroll wheel, move the wheel up to scroll up and down to scroll down. If you scroll so far that no objects are in the view, select **Fit Drawing** from the **View** drop-down on the schematic editor top menu bar (or press **Ctrl+D**) to re-center the entire design, resized to fill the window.

Setting Schematic Editor Options

These options are set under **Schematic Editor** in the **Options** dialog box.

To set **Schematic Editor** options:

1. Click **Tools>Options>General Options**.
2. Click **Schematic Editor**.
3. Make your changes on the appropriate panels:
   - General
   - Fonts
   - Colors
   - Wiring
   - Multiple Placement
   - Symbol Editor
4. Click **OK**.

Schematic Editor Options: General

These options are set in the **General** panel under **Schematic Editor** in the **Options** dialog.

- **Symbol Graphics** – specifies the graphic symbol style (**IEEE** or **Traditional**) to be used on schematics. Each style contains two or more active levels for the graphic objects that comprise a symbol.
- **SubCircuit Pin Spacing** – specifies the pin spacing in grid units for pins on a subcircuit component symbol. The default value is 1 grid unit.
- **Show Pin Labels** – specifies that pin labels are to be displayed in the **Schematic Editor**.
- **Property Display Angle Follows Symbol Angle** – specifies that displayed text properties rotate when the associated component is rotated. Disabled by default.
- **Update Property Display on Definition Update** – specifies that when component symbol definitions are updated, the displayed properties of instances of the updated component on the schematic are automatically updated. Enabled by default. If not enabled, only the symbol graphics are updated.
- **Auto Scroll when close to edges** – specifies that the **Schematic Editor** display will scroll automatically when the cursor is placed close to the edge of the editor window.
- **Show advanced property data** – turns on the display of less often used property tabs, such as Symbol and General for components.
- **Zoom with mouse wheel** – controls the behavior of the mouse wheel on mice so equipped. When enabled, you can zoom while moving the mouse wheel up (zoom in) or

down (zoom out); and scroll vertically by holding down the **Shift** key while moving the mouse wheel or scroll horizontally by holding down the **Ctrl** key while moving the mouse wheel. When disabled, you can scroll vertically with the mouse wheel, and zoom with the mouse wheel by pressing the **Shift** key

- **Net name display** – sets the net name property display distance from the net.
- **Symbol Scaling Factor** – Third-party vendors often follow a different symbol dimensioning system than that used in Maxwell. Symbol sizes that look appropriate in a third-party application may not be appropriate for Maxwell. To mitigate such issues, the Symbol Scaling factor can be used during import to scale the incoming symbol graphics for symbol formats such as SVG by the specified amount. Similarly, when exporting, the symbol can be rescaled to the original dimensions.
- **Selection Colors** – sets the colors of the first and subsequent objects as they are selected on a schematic.

Schematic Editor Options: Fonts

These options are set in the **Fonts** panel under **Schematic Editor** in the **Options** dialog.

Set the **Font Name** (from a drop-down selection list) and **Size** for text used on schematics on the **Fonts** tab of the **Schematic Editor Options** dialog box. A **Sample Text** display window shows the appearance of the specified font and size. **Apply this font to all property displays in the active schematic** specifies that all displays in the schematic editor are to use the font style and size you selected.

Schematic Editor Options: Colors

These options are set in the **Color** panel under **Schematic Editor** in the **Options** dialog.

The **Schematic Objects** panel allows you to set the color for each **Object Type**.

- **Object Type** – a drop-down menu used to select the object type whose colors you want to modify. Types are: **Components**, **Interface Ports**, **Page Ports**, **Global Ports**, **Grounds**, **Graphic Items**, **Wires**, **Buses, Title Blocks, Page Borders,** and **Page Border Text**.
- **Set Color** – implements the color you have selected for the object type.
- **Clear Color** – resets the color of the selected object type to its default value.

The **Individual Definitions** panel lets you set the color for specified **Components** and **Wire Domains**.

- **Component Name** – is used to identify the component whose colors you want to modify. (Visible only when the **Components Object Type** is selected.)
- **Add** – allows you to add a component whose colors you want to define. (Visible only when the **Components Object Type** is selected.)
- **Wire Domain** – a drop-down menu used to select the wire domain whose color you want to modify. Each Wire Domain is assigned a unique color by default. For Maxwell, the Domain is: **Conservative - Electrical**.

  - **Set Color**– implements the color you have selected for the specified component. All future instances of the component added to the active schematic window display will possess the selected color.
  - **Clear Color**– resets the color of the component to its default value.
- **Apply these color settings to the current schematic** – applies the color settings for the selected **Object Type** to objects of that type in the active schematic window.

Schematic Editor Options: Wiring Tab

The following options are set on the **Wiring** panel under **Schematic Editor** in the **Options** dialog box.

Connectivity Options:

- **Show Merge Wire dialog before combining wires** – displays the **Merge Wire** dialog box before combining wires to allow for changes to the merge configuration.
- **Show Split Wire dialog before separating wires** – displays the **Split Wire** dialog box before separating wires to allow for changes to the split configuration
- **When renaming a physically separate piece of wire** – offers two choices:
  – **Split into a different net**
  – **Keep connected (name all pieces the same)**
- **Show GlobalPort Disconnect dialog before separating** – displays the **GlobalPort Disconnect** dialog before separating wires to allow for changes to the global port configuration.

Schematic Editor Options: Multiple Placement

Multiple placement allows you to place several objects of the same type in the schematic using repeated mouse clicks. These options are set in the **Multiple Placement**panel under **Schematic Editor** in the **Options** dialog.

Check the desired objects in the **Items for multiple placement** list.

- **Components**
- **Interface Ports**
- **Grounds**
- **Page Connectors**
- **Global Ports**

Schematic Editor Options: Symbol Editor

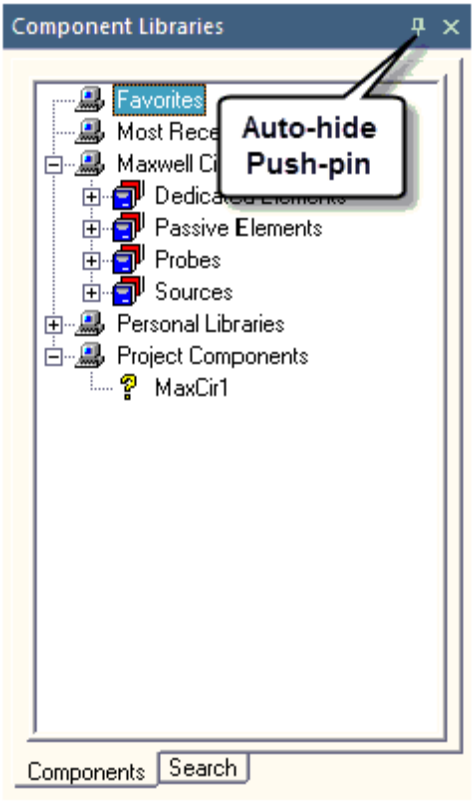
These options are set in the **Symbol Editor** panel under **Schematic Editor** in the **Options** dialog.

- **Adjust Pin Orientation on Drag** – enables symbol pin orientation to be adjusted automatically as pins are dragged to the desired side of a symbol in the symbol editor.

The Component Libraries Window

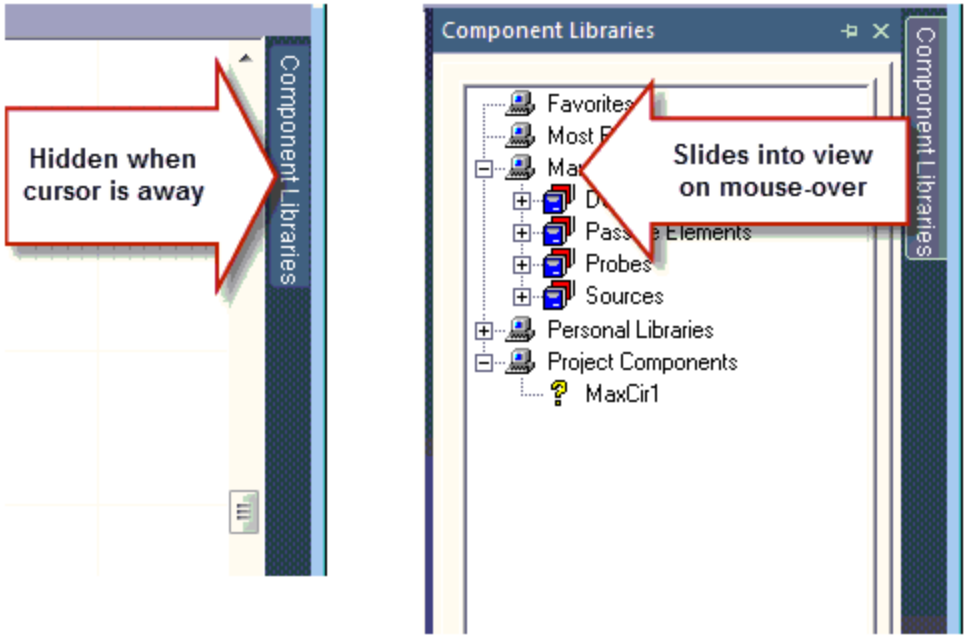
The Component Libraries window is a dockable window that allows you to select and search for schematic components that are available in the installed libraries. The Component Libraries window contains a Components tab and a Search tab.

A typical Component Libraries window is shown below. Your actual window contents will vary with the libraries installed and with use.



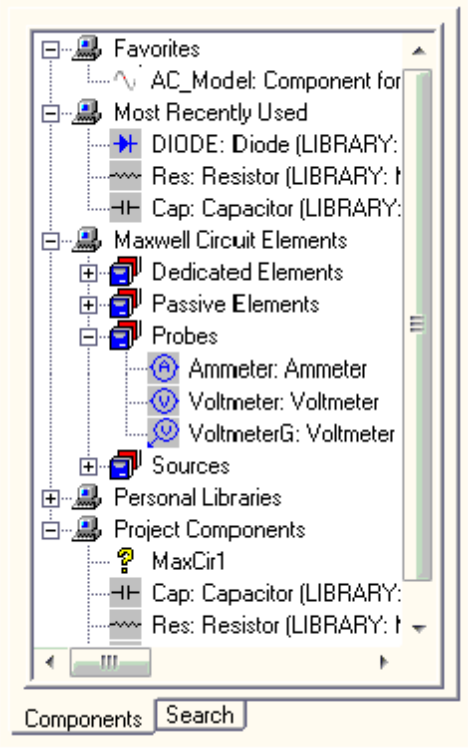
Auto-Hide Feature

Clicking the auto-hide push-pin in the upper right corner of the Component Libraries window toggles the auto-hide feature “on” and “off.” When auto-hide is enabled, the Component Libraries window slides out of sight and a Component Libraries “mouse-over button” appears on the edge of the main window. The Component Libraries window remains hidden until you move the mouse cursor over the mouse-over button. The Component Libraries window then slides into view for use; then hides again when the cursor moves out of the Component Libraries window.



The Components Tab

The **Components** tab in the **Component Libraries** window allows you to browse and select schematic components from the available elements libraries. The installed libraries are listed alphabetically in hierarchical folders grouped by component function. For example, the **Dedicated Elements** library expands as shown.



Individual elements are designated by icons next to the element abbreviation and name.

Right-clicking a component opens a shortcut menu where you can choose to:

- add the component to the Favorites folder
- place the component on a schematic
- view component help

The **Component** tab includes a **Favorites** folder and a **Most Recently Used** folder. These folders help provide quick access to components. Elements used in the current project are listed under **Project Components**.

Using the Favorites and Most Recently Used Folders

To add a library element as a favorite:

1. Select the desired element and right-click to display the shortcut menu.
2. Select **Add to Favorites** to place the component in the **Favorites** folder. When a component is in the **Favorites** folder, you can select, place, edit, view the component help, and load its example project (if one exists) from there.

To remove an element from the **Favorites** folder:

1. Select an element and right click to display the short cut menu.
2. Click **Remove from Favorites**.

The **Most Recently Used** folder contains a list of the most recently used elements. For convenience, these elements can be selected and placed from this folder. The number of elements in the folder is controlled by the **Tools>Options>General Options** dialog under the **Component Libraries Options** tab. The default number of elements is 10.

Help for Components

To launch help for a component from the **Components** tab of the **Component Libraries** window:

1. Right-click its icon.
2. Click **View Component Help**.

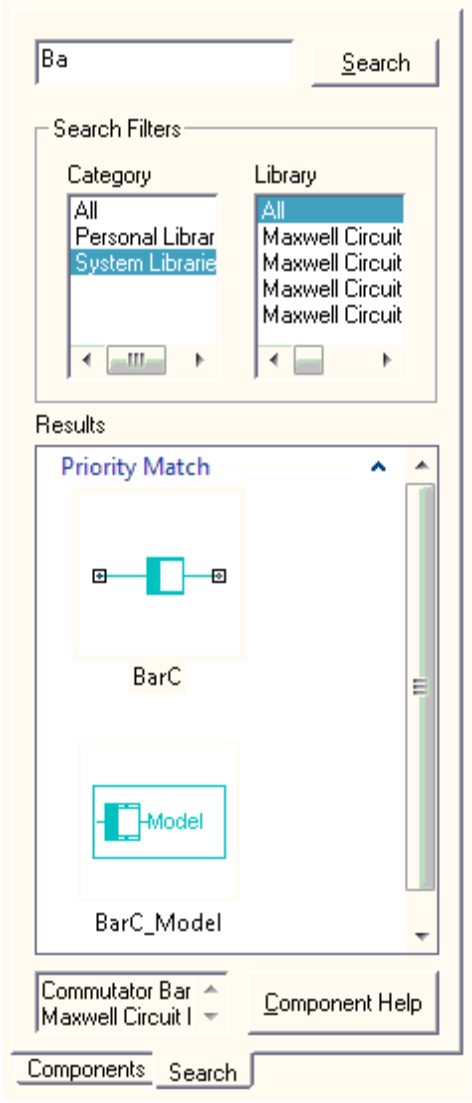
Placing Components on a Schematic

To place a component, do one of the following:

- Double-click its icon.
- Select, drag, and drop a component in the schematic editor.
- Right-click its icon, and then select **Place Component**.

The Search Tab

The **Search** tab allows you to enter full or partial words (case-insensitive) present in the names or descriptions of components in the search field. The search is done dynamically as you type. You can also restrict the search by select the categories and libraries you want to include in the search. Use the **Ctrl** key to select multiple categories or libraries.



For users running Microsoft® Windows Vista® or later, results appear in the **Results** panel as symbols representing the components matched by the search, and arranged in groups, depending on how well a search string matches the component. Generally, components which match the search string exactly appear in the **Best Match** group. The Match groupings do not appear in Windows® versions older than Vista®. Instead the results are simply presented sorted by priority.

When you select a particular component you can see its description and library path in the text box at the bottom of the tab. You can either drag and drop the desired component from the **Results** panel onto the schematic, or double-click to attach it to the mouse. Use the **Component Help** button to view help information for components.

Copying and Pasting Properties

You can copy and paste properties for primitive drawing elements (graphical objects) and components.

Primitive drawing elements (graphical objects)

You can copy and paste the common properties of the following primitive drawing elements (graphical objects):

- Arcs
- Circles
- Lines
- Rectangles
- Polygons
- Text
- Images

Components

Properties are copied only when property names match.

Parameters, and property displays are copied along with properties and their attributes.

Note	Property displays are copied to the same location as the source if they are on the left, bottom, right, top or center. Custom location property displays are copied to the default bottom location.

Related Topics

To Copy and Paste Selected Properties for Components

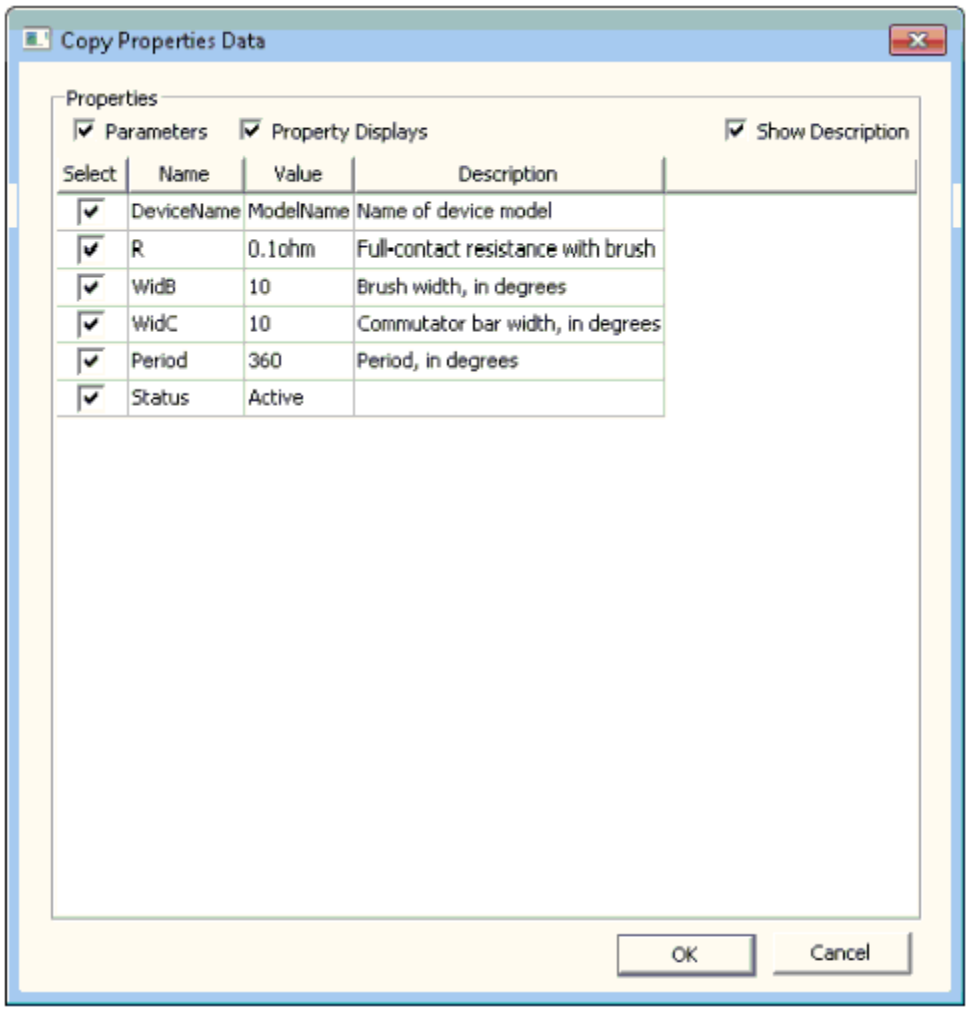
To Copy and Paste Common Properties for Primitive Drawing Elements

To Copy and Paste Selected Properties for Components

To copy and paste selected properties from one component to one or more components:

1. Select the component you want to copy from.
2. Use one of the following methods to access Copy Data:
   - Click **Edit** > **Copy Data**.
   - Right-click the object and select **Copy Data**.
   - Press **Ctrl+Shift+C**.
3. Select the components you want to paste to.
4. Paste the properties by using one of the following methods:
   - Select **Edit** > **Paste Data**.
   - Right-click the component and select **Paste Data.**
   - Press **Ctrl+Shift+V**.

   The **Copy Properties Data** window is displayed.



5. Do one of the following to select the properties to copy.
   - Select one or more of the following check boxes: Parameters, or Property Displays.
   - Click **Select** in the first column to toggle between selecting all and clearing all.
   - Click the **Select** check box for the property. If you do not want to include the property, clear the **Select** check box.

   Click the **Show Description** check box to display the descriptions for each property.
6. Click **OK**. The properties are pasted into the component.

Related Topics

Copying and Pasting Properties

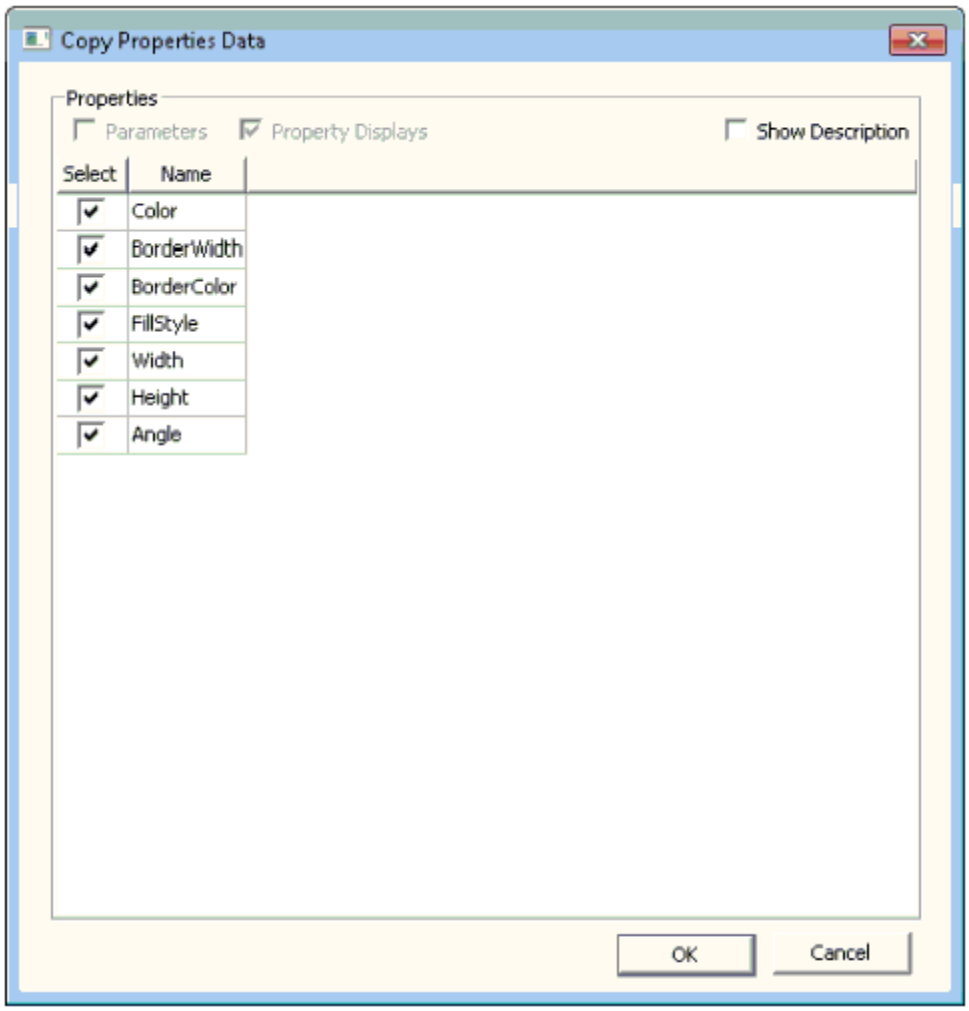
To Copy and Paste Common Properties for Primitive Drawing Elements

To Copy and Paste Common Properties for Primitive Drawing Elements

To copy and paste the common properties from one primitive drawing element (graphical object) to another:

1. Select the object you want to copy from.
2. Use one of the following methods to access Copy Data:
   - Click **Edit** > **Copy Data**.
   - Right-click the object and select **Copy Data**.
   - Press **Ctrl+Shift+C**.
3. Select the object you want to paste to.
4. Paste the properties by using one of the following methods:
   - Select **Edit** > **Paste Data**.
   - Right-click the component and select **Paste Data.**
   - Press **Ctrl+Shift+V**.

   The **Copy Properties Data** window is displayed.



5. In the **Properties** pane, keep the default selections or clear the check boxes for the properties you do not want to copy.

   Click the **Show Description** check box to display the description for each property.
6. Click **OK**. The properties are pasted into the object.

Related Properties

Copying and Pasting Properties

To Copy and Paste Selected Properties for Components

Components

A number of components are available in the Circuit simulator.

To view and select components, click the **Components** tab in the **Component Libraries Window**. To expand a component subgroup, double-click its book icon. To read about a specific component, right-click its icon in the tree and select **View Component Help**.

The following types of elements are available in Maxwell Circuit Editor:

- Dedicated Elements
- Passive Elements
- Probes
- Sources

Once components are placed in the schematic for a project, they appear in the project tree beneath the **Project Components** branch. The most recently placed components also appear under the **Most Recently Used** branch in the project tree. You can also set **Favorites** that can be accessed from the project tree.

Related Topics

Placing Components in the Maxwell Circuit Editor Schematic

Assigning Component Properties in Maxwell Circuit Editor

Dedicated Elements

The following dedicated elements are available in the Maxwell Circuit Editor project tree:

- AC_Model: Component for identifying windings with AC currents
- BarC: Commutator Bar
- BarC_Model: Model Data for Commutator Bar
- ECE3_Model: ECE 3-Phase Model
- ECE6_MODEL: ECE 6-Phase Model
- ECEF_MODEL: ECE Frequency Model
- ECEIM_MODEL: ECE Induction Motor Model
- ECEL_Model: ECE Linear Motion Model
- ECER_Model: ECE Rotation Model
- ECESRM_Model: ECE Switched Reluctance Motor Model
- ECET_Model: ECE Transformer Model
- ECEW_Model: ECE Winding Model
- Winding: Winding

The text before the colon (**:**) represents the component name and can be changed in the **Properties** window once the component is placed in the schematic.

Related Topics

Assigning Component Properties in Maxwell Circuit Editor

AC_Model

The AC_model element is used to decrease the number of electric periods required to reach steady state when feeding windings of inductive loads with voltage sources in the external circuit

editor. It assumes that the average current over one electric period in all windings defined in the AC_model is zero.

After each electric period, the initial current value of the windings is automatically manipulated to compensate for the offset current of the previous electric period. This allows you to modify currents for AC windings at the beginning of each period without interrupting the transient process. If the offset current of the last period is below a defined tolerance, no manipulation of the phase currents is triggered.

To change the name of the model placed on the sheet, click the symbol on the sheet, and change the name of the component in the property window (Value field in the DeviceName line, with the Param Values tab selected).

The AC_Model contains the following parameters:

- DeviceName, which identifies the name of device model.
- Windings, which is the AC winding list separated by commas. For example, PhA, PhB, PhC. Setting this parameter to “All” will include all windings.
- Frequency, which shows the frequency of the AC currents in Hz.
- Tolerance, which is the ratio of the DC component to the AC amplitude that defines the threshold value triggering a manipulation of the initial current for the subsequent electric period.

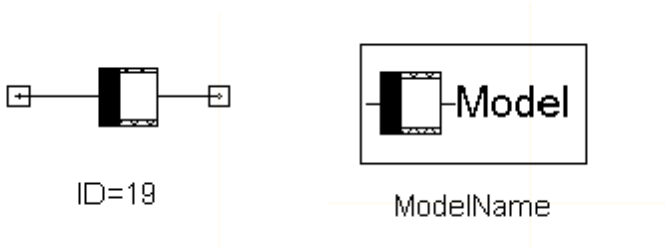
> **Note:** For Maxwell 2D/3D designs of AC machines, RMxprt automatically enables an alternating flux (AF) model in the voltage definition for each winding to reduce transient solution time to reach steady state when the Maxwell 2D or 3D model is created. Also refer to *Using the AF (alternating flux) model* for information on using the AF model to fast reach steady state for transient simulations whose excitation is the Sinusoidal Voltage Source component.

Commutator Bar and Commutator Bar Model

The commutator bar element is intended to be used for the motor model with a commutator. This element models the variable (periodic) contact resistance between the brush and the commutator bars, as well as the switching (commutation of the current) that occurs when the brush makes contact with the two adjacent commutator bars.

The element itself must always be used together with the corresponding commutator bar model. The commutator bar model can be dropped on the sheet anywhere and needs no connections.

Only the commutator bar element itself should be connected as required by the application. The commutator bar elements need to reference the applicable commutator model.



Once the commutator bar element has been dropped on the sheet, you can double-click it to access the properties (make sure the **Parameter Values** tab is selected). Specify the applicable commutator bar model name in the MOD line and also the Lag parameter in degrees. Lag identifies the angle the commutator bar has to rotate from TIME = 0 in the chosen sense of rotation until it is perfectly aligned with the brush. By defa1ult, the element ID and lagging angle in degrees are displayed next to the element.

The commutator bar model needs to be dropped on the circuit sheet. It is unique for every commutator bar element. The commutator bar model contains the following parameters:

- Model name that has to be referenced by all the commutator bar elements;
- R, the full contact resistance between brush and commutator bar, regardless of which of the two is wider;
- WidB is the brush width in mechanical degrees;
- WidC is the commutator bar width in mechanical degrees (does not include the insulation between two adjacent bars);
- Period is the angular periodicity of the positive (or negative) brushes; use 360 for a two pole machine, 180 for a four pole machine with lap winding, etc.

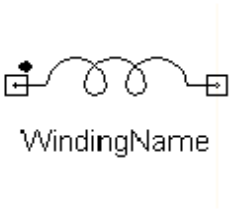
Related Topics

An Application of the Commutating Bar Element.

Winding

The winding element is used in the Maxwell Circuit Editor to create the necessary connection between the finite element model (the type of solution that supports the concept of winding, such as the transient type of analysis, with or without motion) and the driving circuits. It is necessary that the name(s) assigned for the winding(s) in the finite element model are matched exactly in the driving circuit created in Maxwell Circuit Editor. Windings can be placed on the design sheet at any moment while you are creating the circuitry to be used to drive the finite element windings.

To change the name of the winding placed on the sheet, click the winding symbol on the sheet, and change the name of the component in the property window (Value field in the DeviceName line, with the Param Values tab selected).

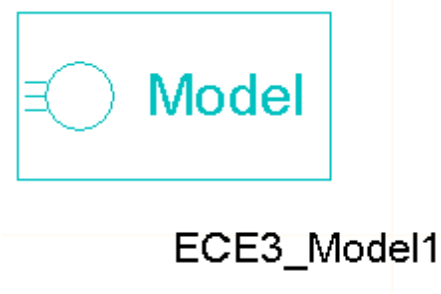


Note	The dot next to the winding symbol is used as the positive reference for the initial current (positive current is oriented from the "dotted" terminal toward to "un-dotted" terminal of the winding, through the winding).

ECE3_Model

The ECE3 model enables you to create an ECE model from Maxwell Transient solutions. Compared with the procedure for creating ECE model from Maxwell Magnetostatic solutions, creating an ECE model directly from Maxwell Transient solutions takes advantage of winding and motion setups in Maxwell Transient.

The ECE3 model is used to set up the sweeping of currents in three-phase windings. To change the name of the model placed on the sheet, click the symbol on the sheet, and change the name of the component in the property window (Value field in the DeviceName line, with the Parameter Values tab selected).



The ECE3_Model contains the following parameters:

- DeviceName – the name of the device model with default name ECE3_Model1.
- Windings – specifies the comma-separated name list of three-phase windings whose currents are to be swept.
- CurrentSweeps – specifies the current amplitude sweep list. The format is: "v1, v2, (dv1, n1), v3, …", where v1, v2, v3 specify individual sweeping values, and (dv1, n1) specify a sweep range with dv1 being the increment, and n1 the number of increments. For example, a list of "(0.5A, 5)" is equivalent to "0, 0.5A, 1A, 1.5A, 2A, 2.5A".
- PhAngIntervals – specifies the sweep types, or the number of phase-angle sweep intervals. For three-phase windings, only two independent variables are swept to derive

three-phase currents, instead of directly sweeping three-phase currents. The two independent variables could be DQ currents (I_d, I_q), two-phase currents (I_α, I_β), or current amplitude and phase angle (I_m,β).

PhAngIntervals = 0, 1, or 2, is defined for the DQ current sweep type, and 3 for the two-phase current sweep type.

- When PhAngIntervals = 0, both I_d and I_q are swept based on the original setup of CurrentSweeps, and the look-up table is extended according to odd-symmetric conditions in a post-processing step.
- When PhAngIntervals = 1, the I_d sweep is extended symmetrically to sample negative and positive current values. In contrast, the I_q sweep is based on the original setup of CurrentSweeps and extended in the look-up table according to odd-symmetric conditions in a post-processing step.
- When PhAngIntervals = 2, both I_d and I_q sweeps are extended symmetrically from negative to positive, sampling positive and negative values for both, I_d and I_q. Accordingly, the computed look-up table does not need to be extended in post-processing.
- When PhAngIntervals = 3, both I_α and I_β sweeps are extended symmetrically from negative to positive, and the computed look-up table is not extended in post-processing.
- When PhAngIntervals ≥ 12, the current amplitude and phase angle sweep type are defined. The current amplitude I_m is swept based on the setup of CurrentSweeps, and the phase angle is swept from 0 to 360 electrical degrees with the increment $\Delta\beta = 360°/(\text{PhAngIntervals})$.

- Res – phase winding resistance (available when Show Hidden is checked. Default value is 1mΩ).
- IndE – phase end-leakage inductance (available when Show Hidden is checked. Default value is 1μH).

An ECE3_Model can be used individually to create the ECE model of a three-phase inductor, or combined with the ECER_Model for three-phase PM machines. It can also be combined with the ECER_Model and ECEW_Model for winding excited three-phase synchronous machines. An ECE3_Model with DQ current sweep type must always be used with an ECER_Model.

For ECE model creation, the setup of Stop Time and Time step are ignored. At Maxwell run time, the circuit simulator assigns currents for all windings step-by-step based on the lists of current sweeps until all sweeps are finished. An ECE model is then created automatically. You can import this ECE model in Twin Builder via the **Twin Builder>SubCircuit>Maxwell Component>Add Equivalent Circuit** menu command. For more information on coupling Maxwell designs to Twin Builder components, refer to the *Maxwell Equivalent Circuit Component* topic in the Twin Builder Help.

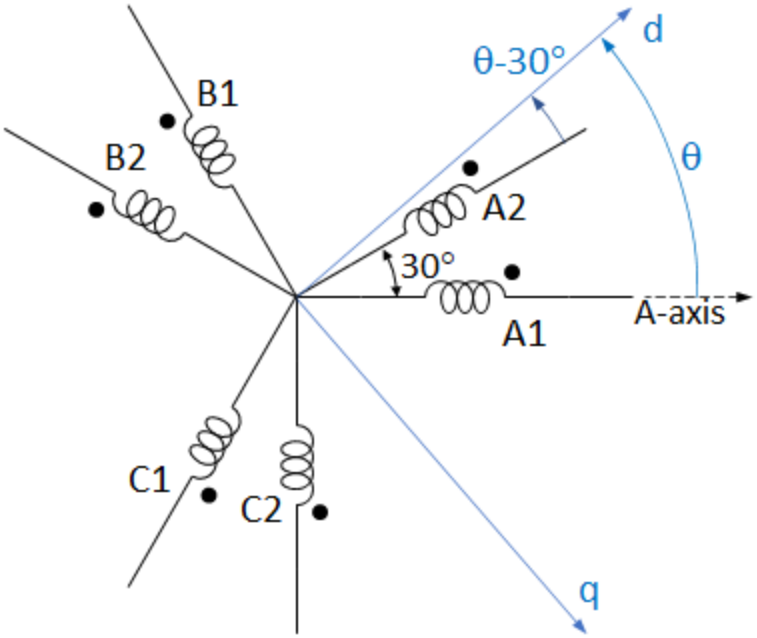
Note	You can add an ECE3_Model to an existing external circuit without deleting the original circuit. In such cases, the original circuit is ignored and can be kept for future use. If you want to recover the original circuit for a transient simulation, instead of ECE model creation, you just need to deactivate each ECE related model by right-clicking the model and selecting Deactivate in the short-cut menu.

For more information on coupling Maxwell designs to Twin Builder components, refer to the *Coupling Maxwell to a Twin Builder Component* topic in the Maxwell Help

ECE6_Model

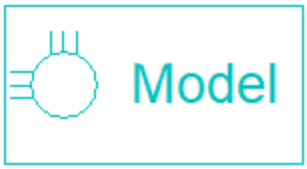
The ECE6 model enables you to create an ECE model from Maxwell Transient solutions. Compared with the procedure for creating ECE model from Maxwell Magnetostatic solutions, creating an ECE model directly from Maxwell Transient solutions takes advantage of winding and motion setups in Maxwell Transient.

The ECE6 model is used to set up the sweeping of currents in a six-phase winding, which is described as dual three-phase windings. These two three-phase windings are shifted by 30 electrical degrees. The figure below shows the set-2 three-phase windings lagging behind the set-1 three-phase windings by 30 electrical degrees, and dq axes in rotor with the q-axis lagging behind the d-axis by 90 electrical degrees.



We define the rotor initial position with θ = 0 as the position where the rotor d-axis aligns with the axis of the phase A1 winding. That is, the phase A1 gets maximum flux linkage when θ = 0.

To change the name of the model placed on the sheet, click the symbol on the sheet, and change the name of the component in the property window (Value field in the DeviceName line, with the Parameter Values tab selected).



ECE6_Model1

The ECE6_Model contains the following parameters:

- DeviceName – the name of the device model with default name ECE6_Model1.
- Windings – specify winding names in the order of arrangement. Windings are separated by commas within each three-phase set, and by semicolon between the two three-phase sets. Please notice that the winding list always starts with the leading set and ends with the lagging set. For the phase windings as shown in the figure above, the winding list is specified as:

  A1 B1, C1; A2, B2, C2
- CurrentSweeps – to specify sweeping values for d- and q-axis currents, Id and Iq, respectively, you only need to list 0 and all positive sweeping currents. All negative currents will be swept, or extended, symmetrically from the specified positive sweeping values. For example: 0, 1, 10
- PhAngIntervals – specify the type to extend the sweeping list and look-up table. Valid values are: 0, 1, or 2 only. Value 0 specifies the type that both Id and Iq are post-extended; Value 1 specifies the type that Id is pre-extended, and Iq is post-extended; Value 2 specifies the type that both Id and Iq are pre-extended. Here, “pre-extended” means the sweeping list for a current is symmetrically extended from the sweeping values specified in CurrentSweeps, and “post-extended” means the look-up table for a current, which is derived from the sweeping values specified in CurrentSweeps, is extended based on even or odd symmetric conditions. In a post extension, the symmetric conditions are applicable only for fundamental fields. Therefore, it is not necessary to consider all harmonics in the look-up table for post-extended variables. In such a case, all harmonic components will be filtered by using the average values over various rotor positions for all dq0 flux linkages and torque to reduce the length of the look-up table.
- Res – phase winding resistance (available when Show Hidden is checked).
- IndE – phase end-leakage inductance (available when Show Hidden is checked).

Transformation between the six-phase and dq0 systems

The current transformation from dq0 system to set-1 ABC system is:

$$\begin{bmatrix} I_{A1} \\ I_{B1} \\ I_{C1} \end{bmatrix} = \mathbf{C}(\theta) \cdot \begin{bmatrix} I_{d1} \\ I_{q1} \\ I_{01} \end{bmatrix}$$

where

$$\mathbf{C}(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 1 \\ \cos\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta - \frac{2\pi}{3}\right) & 1 \\ \cos\left(\theta - \frac{4\pi}{3}\right) & \sin\left(\theta - \frac{4\pi}{3}\right) & 1 \end{bmatrix}$$

Since the set-2 winding lags behind the set-1 winding by 30 electrical degrees, the current transformation from dq0 system to set-2 ABC system is:

$$\begin{bmatrix} I_{A2} \\ I_{B2} \\ I_{C2} \end{bmatrix} = \mathbf{C}\left(\theta - \frac{\pi}{6}\right) \cdot \begin{bmatrix} I_{d2} \\ I_{q2} \\ I_{02} \end{bmatrix}$$

The contribution of 0-axis currents (I_{01}, I_{02}) to winding flux linkages can be considered by zero-inductance L_0, which is usually assumed to be constant. Therefore, it is not necessary to sweep 0-axis currents. In general, to cover all possible operation cases, the remaining four independent currents (I_{d1}, I_{q1}, I_{d2}, I_{q2}) need to be swept. However, in most applications, to get the benefits of 6-phase windings, these dual three-phase windings are operated with balanced currents, that is, in the winding set lagging behind by 30 electrical degrees in space, the currents lag behind by 30 degrees in time. In such cases, the dq currents on both winding sets are the same. Therefore, to save FEA simulation time, we sweep only two independent currents (I_d, I_q) during FEA simulation, and assign

$$\begin{cases} I_{d1} = I_{d2} = I_d \\ I_{q1} = I_{q2} = I_q \end{cases}$$

During circuit simulation, we measure dual three-phase currents (I_{A1}, I_{B1}, I_{C1}) and (I_{A2}, I_{B2}, I_{C2}), and transform them to dq currents by

$$\begin{bmatrix} I_{d1} \\ I_{q1} \\ I_{01} \end{bmatrix} = \mathbf{C}^{-1}(\theta) \cdot \begin{bmatrix} I_{A1} \\ I_{B1} \\ I_{C1} \end{bmatrix}$$

and

$$\begin{bmatrix} I_{d2} \\ I_{q2} \\ I_{02} \end{bmatrix} = \mathbf{C}^{-1}\left(\theta - \frac{\pi}{6}\right) \cdot \begin{bmatrix} I_{A2} \\ I_{B2} \\ I_{C2} \end{bmatrix}$$

where

$$\mathbf{C}^{-1}(\theta) = \frac{2}{3} \begin{bmatrix} \cos(\theta) & \cos\left(\theta - \frac{2\pi}{3}\right) & \cos\left(\theta - \frac{4\pi}{3}\right) \\ \sin(\theta) & \sin\left(\theta - \frac{2\pi}{3}\right) & \sin\left(\theta - \frac{4\pi}{3}\right) \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

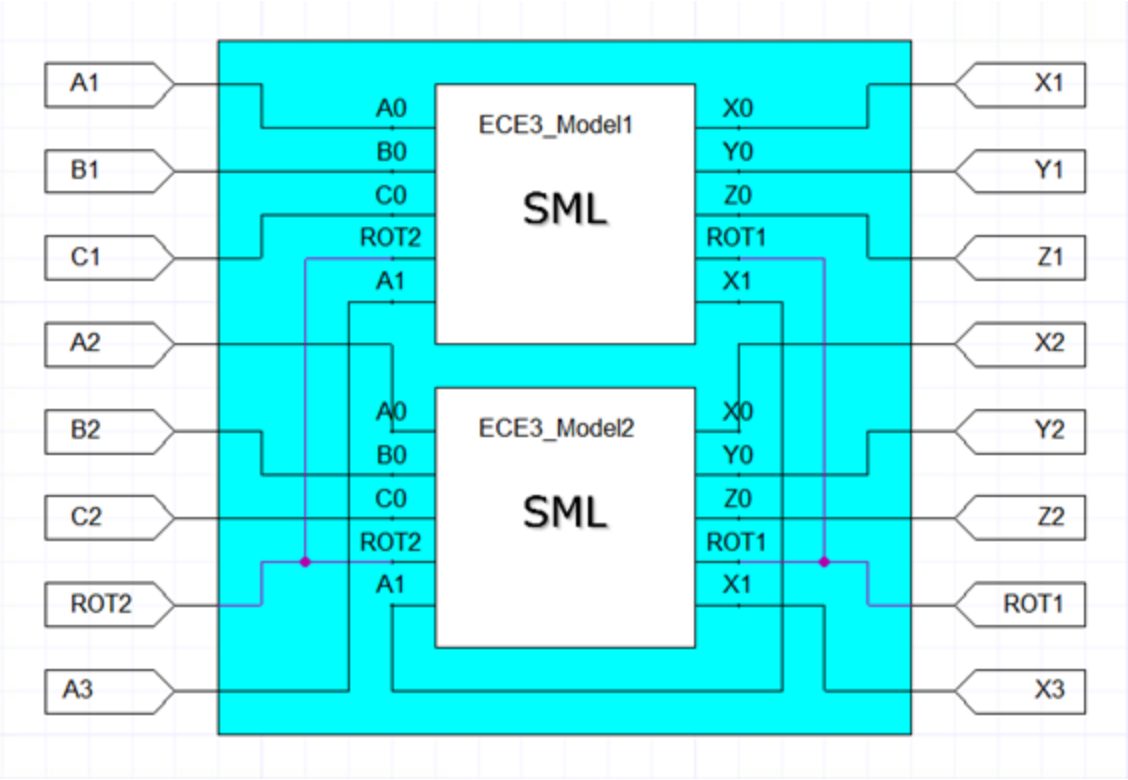
The dq currents used in the look-up table are:

$$\begin{cases} I_d = (I_{d1} + I_{d2})/2 \\ I_q = (I_{q1} + I_{q2})/2 \end{cases}$$

Since the two sets of three-phase windings have the same turns, the fundamental field produced by the average dq currents will be the same as that produced by the real dq currents. Therefore, for unbalanced operations, replacing the real dq currents (I_{d1}, I_{q1}) and (I_{d2}, I_{q2}) in the two winding sets by the average dq currents (I_d, I_q) will not cause noticeable errors.

Circuit Model and Look-up Table

The circuit model for 6-phase machines consists of two sub-circuits of 3-phase machines with the dq currents of the other sub-circuit being the input arguments, as shown below, where the initial position, an input argument of the sub-circuit model, of ECE3_Model2 is set 30 electrical degrees lagging behind ECE3_Model1. The mechanical ports represented by (ROT1, ROT2) of the two sub-circuit are connected in parallel, and the torque values in the look-up table are a half of the FEA solutions for each sweep. The field winding, if it exists, represented by (A3, X3) in the circuit model, is obtained by series connection of the field windings, represented by (A1, X1), of the two sub-circuits.



The flux linkage values in the look-up table of the field winding in the sub-circuit are a half of the FEA solutions. The input dq currents of the other sub-circuit are used to get the average dq currents as the input of the look-up table. The following is an example of the data structure of the look-up table of a 6-phase machine with wound field.

```
B_BasicData
  Version   1.0
  Poles     4
  Lld  0.000888929
E_BasicData

B_PhaseImp 7
    PhaseA    1.0000000000e-03    1.0000000000e-06
    PhaseB    1.0000000000e-03    1.0000000000e-06
    PhaseC    1.0000000000e-03    1.0000000000e-06
   PhaseA1    1.0000000000e-03    1.0000000000e-06
   PhaseB1    1.0000000000e-03    1.0000000000e-06
   PhaseC1    1.0000000000e-03    1.0000000000e-06
     Field    1.0000000000e-03    1.0000000000e-06
E_PhaseImp

B_Sweepings
  Id_Iq     (7:  -10 -5 -1 0 1 5 10)
            (7:  -10 -5 -1 0 1 5 10)
  Iw        (5:  0 4 5 6 10)
  Rotate    (49:  0 1.25 2.5 ... 60)
E_Sweepings

B_OutputMatrix DQ0
        0   -2.0359505219e-01   -1.1958557789e-01    6.8140161295e-03   -3.7702990147e-01   -3.1186012134e+00
        1   -2.0314518063e-01   -1.1829390416e-01    7.9496955252e-03   -3.7825548839e-01   -3.1680201040e+00
    ......
    12004    2.5756484736e-01    8.1949917793e-02   -1.0324170975e-03    5.6583790148e-01   -5.4424987350e+00
E_OutputMatrix
```

The sub-circuit data include basic data, specified resistances and leakage inductances of all windings, value list for all controlling variables, and value list for all controlled variables of al sweeps.

The basic data are listed between “B_BasicData” and “E_BasicData”, which include the version number, the number of poles, and d- and q-axis leakage inductance between the 2-set 3-phase windings.

The data between “B_PhaseImp” and “E_PhaseImp” list the winding resistances and end leakage inductances, which are not included in 2D FEA results, for each winding. The winding resistances and end leakage inductances are obtained from specified parameters Res and IndE of the ECE6_Model, as well as ECEW_Model if it is included.

The data between “B_Sweeping” and “E_Sweeping” provide all sweeping variables, and their sweeping data lists. Each sweeping variable, corresponding to one ECE sweeping model, starts with a keyword and follows one or two sweeping data list. The sweeping data list is in the format of (n: v0 v1 …) representing the number of sweeps and sweeping values. Keyword “Id_Iq” means the dq current sweep, “Iw” represents wound field current sweep, and “Rotate” indicates the rotor position sweep.

The data between “B_OutputMaxtrix” and “E_OutputMatrix” lists solutions of all sweeps. The first column is the sweeping index from 0 to n-1, here n is the total number of sweeps. The total number of sweeps is 7x7x5x49 = 12005. The 2nd to 4th columns list the dq0 flux linkages of all sweeps for the present sub-circuit. If the field winding exists, the 5th column lists its flux linkage. The last column list torque values of all sweeps.

From FEA solutions, we get results of flux linkages of all windings. The dq0 flux linkages can be obtained from

$$\begin{bmatrix} \lambda_{d1} \\ \lambda_{q1} \\ \lambda_{01} \end{bmatrix} = \mathbf{C}^{-1}(\theta) \cdot \begin{bmatrix} \lambda_{A1} \\ \lambda_{B1} \\ \lambda_{C1} \end{bmatrix}$$

and

$$\begin{bmatrix} \lambda_{d2} \\ \lambda_{q2} \\ \lambda_{02} \end{bmatrix} = \mathbf{C}^{-1}\left(\theta - \frac{\pi}{6}\right) \cdot \begin{bmatrix} \lambda_{A2} \\ \lambda_{B2} \\ \lambda_{C2} \end{bmatrix}$$

Since set-2 windings lag behind set-1 windings by 30 electrical degrees, the ABC flux linkages of the set-2 windings will lag behind those of the set-1 windings by 30 degrees. As the result, the dq0 flux linkages satisfy

$$\begin{cases} \lambda_{d1}\left(\theta + 30^{\circ}\right) = \lambda_{d2}\left(\theta\right) \\ \lambda_{q1}\left(\theta + 30^{\circ}\right) = \lambda_{q2}\left(\theta\right) \\ \lambda_{01}\left(\theta + 30^{\circ}\right) = -\lambda_{02}\left(\theta\right) \end{cases}$$

The repeatable period for the d- and q-axis flux linkages is 60 electrical degrees, but that for the 0-axis flux linkage is 120 electrical degrees. Therefore, all dq0 flux linkages must be extended to 120 electrical degrees by

$$\begin{cases} \lambda_{d1}\left(\theta + 60^{\circ}\right) = \lambda_{d1}\left(\theta\right) \\ \lambda_{q1}\left(\theta + 60^{\circ}\right) = \lambda_{q1}\left(\theta\right) \\ \lambda_{01}\left(\theta + 60^{\circ}\right) = -\lambda_{01}\left(\theta\right) \end{cases}$$

The extended dq0 flux linkages in the final look-up table are based on the same dq currents in two winding sets. During the circuit simulation, if the dq currents in the two windings are not the same, the dq flux linkages obtained from the look-up table must be modified as described below. The modified d-axis flux linkage is

$$\lambda_{d1}{}' = \lambda_{Fd} + L_{d1} I_{d1} + M_{d12} I_{d2}$$

$$= \lambda_{d1} + L_{1d}\left(I_{d1} - I_{d2}\right)/2$$

where λ_{Fd} is the flux linkage produced by wound field winding current, or PM, and

$$L_{1d} = L_{d1} - M_{d12}$$

$$\lambda_{d1} = \lambda_{Fd} + L_{d1} I_d + M_{d12} I_d$$

which can be directly obtained from the look-up table. Similarly, the q-axis flux linkage is modified by

$$\lambda_{q1}{}' = \lambda_{q1} + L_{1q}\left(I_{q1} - I_{q2}\right)/2$$

where $L_{1q} = L_{1d}$, and λ_{q1} can be directly obtained from the look-up table. To filter the slot effects, the d- or q-axis leakage inductance L_{1d} can be computed from

$$L_{1d} = \frac{1}{n}\sum_{i=1}^{n} 0.5\left[\left(L_{d1} - M_{d12}\right)_i + \left(L_{q1} - M_{q12}\right)_i\right]$$

where *i* represents the sweeping index.

The d- and q-axis self-inductance and mutual inductance between 2 winding sets are derived from

$$\begin{bmatrix} L_{d1} & 0 & 0 \\ 0 & L_{q1} & 0 \\ 0 & 0 & L_{01} \end{bmatrix} = \mathbf{C}^{-1}(\theta) \cdot \begin{bmatrix} L_{A1} & M_{A1B1} & M_{A1C1} \\ M_{B1A1} & L_{B1} & M_{B1C1} \\ M_{C1A1} & M_{C1B1} & L_{C1} \end{bmatrix} \cdot \mathbf{C}(\theta)$$

and

$$\begin{bmatrix} M_{d12} & 0 & 0 \\ 0 & M_{q12} & 0 \\ 0 & 0 & M_{012} \end{bmatrix} = \mathbf{C}^{-1}(\theta) \cdot \begin{bmatrix} M_{A1A2} & M_{A1B2} & M_{A1C2} \\ M_{B1A2} & M_{B1B2} & M_{B1C2} \\ M_{C1A2} & M_{C1B2} & M_{C1C2} \end{bmatrix} \cdot \mathbf{C}\left(\theta - \frac{\pi}{6}\right)$$

An ECE6_Model should be used with an ECER_Model.

For ECE model creation, the setup of Stop Time and Time step are ignored. At Maxwell run time, the circuit simulator assigns currents for all windings step-by-step based on the lists of current sweeps until all sweeps are finished. An ECE model is then created automatically. You can import this ECE model in Twin Builder via the **Twin Builder>SubCircuit>Maxwell Component>Add Equivalent Circuit** menu command. For more information on coupling Maxwell designs to Twin Builder components, refer to the *Maxwell Equivalent Circuit Component* topic in the Twin Builder Help.

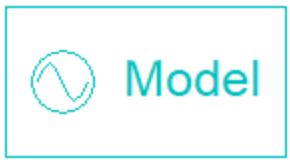
Note	You can add an ECE6_Model to an existing external circuit without deleting the original circuit. In such cases, the original circuit is ignored and can be kept for future use. If you want to recover the original circuit for a transient simulation, instead of ECE model creation, you just need to deactivate each ECE related model by right-clicking the model and selecting Deactivate in the short-cut menu.

For more information on coupling Maxwell designs to Twin Builder components, refer to the *Coupling Maxwell to a Twin Builder Component* topic in the Maxwell Help

ECEF_Model

The ECEF model enables you to create an ECE model from Maxwell Eddy Current solutions.

The ECEF model is used to set up a frequency sweep. To change the name of the model placed on the sheet, click the symbol on the sheet, and change the name of the component in the property window (Value field in the DeviceName line, with the Parameter Values tab selected).



ECEF_Model1

The ECEF_Model contains the following parameters:

- DeviceName –the name of the device model with default name ECEF_Model1.
- FreqSweeps – specifies the frequency sweep list. The format is: "v1, v2, (dv1, n1), v3, …", where v1, v2, v3 specify individual sweeping values, and (dv1, n1) specify a sweep range with dv1 being the increment, and n1 the number of increments. For example, a list of "50Hz, (10Hz, 5)" is equivalent to "50Hz, 60Hz, 70Hz, 80Hz, 90Hz, 100Hz".

An ECEF_Model cannot be used individually. It should be used combined with an ECEIM_Model for induction motors.

At Maxwell run time, the circuit simulator assigns current, slip, as well as frequency if ECEF_ Model is added, to Maxwell Eddy Current solver until all sweeps are finished. An ECE model is created automatically. You can import this ECE model in Twin Builder via the **Twin Builder>SubCircuit>Maxwell Component>Add Equivalent Circuit** menu command. For more information on coupling Maxwell designs to Twin Builder components, refer to the *Maxwell Equivalent Circuit Component* topic in the Twin Builder Help.

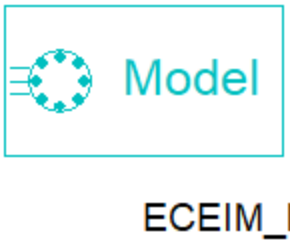
Note	You can add an ECEF_Model to an existing external circuit without deleting the original circuit. In such cases, the original circuit is ignored and can be kept for future use. If you want to recover the original circuit for a simulation, instead of ECE model creation, you just need to deactivate each ECE related model by right-clicking the model and selecting Deactivate in the short-cut menu.

For more information on coupling Maxwell designs to Twin Builder components, refer to the *Coupling Maxwell to a Twin Builder Component* topic in the Maxwell Help

ECEIM_Model

The ECEIM model enables you to create an ECE model of a three-phase induction motor from Maxwell Eddy Current solutions.

To change the name of the model placed on the sheet, click the symbol on the sheet, and change the name of the component in the property window (Value field in the DeviceName line, with the Parameter Values tab selected).



The ECEIM_Model contains the following parameters:

- DeviceName – the name of the device model with default name ECEIM_Model1.
- Windings – specifies the comma-separated name list of three-phase windings whose currents are to be swept.
- CurrentSweeps – specifies lists of sweep currents. The format is: “v1, v2, (dv1, n1), v3, …”, where v1, v2, v3 specify individual sweep values, and (dv1, n1) specify a sweeping range with dv1 being the increment, and n1 the number of increments. For example, a list of “-2.5A, (0.5A, 10)” is equivalent to "-2.5A, -2A, -1.5A, -1A, -0.5A, 0, 0.5A, 1A, 1.5A, 2A, 2.5A", and a list of "(0.5A, 5)" is equivalent to “0, 0.5A, 1A, 1.5A, 2A, 2.5A”.
- SkewAng – specifies the skew angle (in mechanical degrees) of the stator or rotor core for 2D only (available when Show Hidden is checked).
- Res – phase winding resistance (available when Show Hidden is checked).
- IndE – phase end-leakage inductance (available when Show Hidden is checked).

The ECEIM model is used to set up the sweeping of currents in three-phase windings. A slip sweep is added together with the current sweep. The slip sweep is fixed (not user-configurable) with a list of the following 13 values: 0.01, 0.02, 0.04, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0. To simulate the real operation of induction motors at a specified slip with Eddy Current solvers, the rotating rotor is treated as a locked rotor with equivalent rotor resistances and inductances.

An ECEIM_Model can be used individually to create the ECE model of a three-phase induction machine, or combined with the ECEF_Model for sweeping frequency.

At Maxwell run time, the circuit simulator assigns current, slip, as well as frequency if ECEF_ Model is added, to Maxwell Eddy Current solver until all sweeps are finished. An ECE model is then created automatically. You can import this ECE model in Twin Builder via the **Twin Builder>SubCircuit>Maxwell Component>Add Equivalent Circuit** menu command. For more information on coupling Maxwell designs to Twin Builder components, refer to the *Maxwell Equivalent Circuit Component* topic in the Twin Builder Help.

Note	You can add an ECEIM_Model to an existing external circuit without deleting the original circuit. In such cases, the original circuit is ignored and can be kept for future use. If you want to recover the original circuit for a simulation, instead of ECE model creation, you just need to deactivate each ECE related model by right-clicking the model and selecting Deactivate in the short-cut menu.

For more information on coupling Maxwell designs to Twin Builder components, refer to the *Coupling Maxwell to a Twin Builder Component* topic in the Maxwell Help.

Circuit and Parameters

Due to saturation and skin effects, the stator leakage inductance might be nonlinear, and the rotor leakage inductance will be nonlinear and slip dependent. The rotor leakage inductance can be decoupled into two parts, non-linear only and slip dependent only, as explained below.

The stator and rotor non-linear can be expressed as

$$\begin{cases} L_1\left(I_1\right) = L_{10} + L_{1m}/k_{s1}\left(I_1\right) \\ L_2\left(I_1\right) = L_{20} + L_{2m}/k_{s2}\left(I_1\right) \end{cases}$$

where

$$\begin{cases} k_{s1}\left(I_1\right) = \left(I_1/I_{T1}\right)/\left(1 - e^{-I_1/I_{T1}}\right) \\ k_{s2}\left(I_1\right) = \left(I_1/I_{T2}\right)/\left(1 - e^{-I_1/I_{T2}}\right) \end{cases}$$

The slip-dependent rotor resistance and leakage inductance are

$$\begin{cases} R_{2s} = k_r R_{2s0} \\ L_{2s} = k_x L_{2s0} \end{cases}$$

where

$$\begin{cases} k_r\left(\xi\right) = \xi\frac{\sinh(2\xi)+\sin(2\xi)}{\cosh(2\xi)-\cos(2\xi)} \\ k_x\left(\xi\right) = \frac{3}{2\xi}\frac{\sinh(2\xi)-\sin(2\xi)}{\cosh(2\xi)-\cos(2\xi)} \end{cases}$$

with

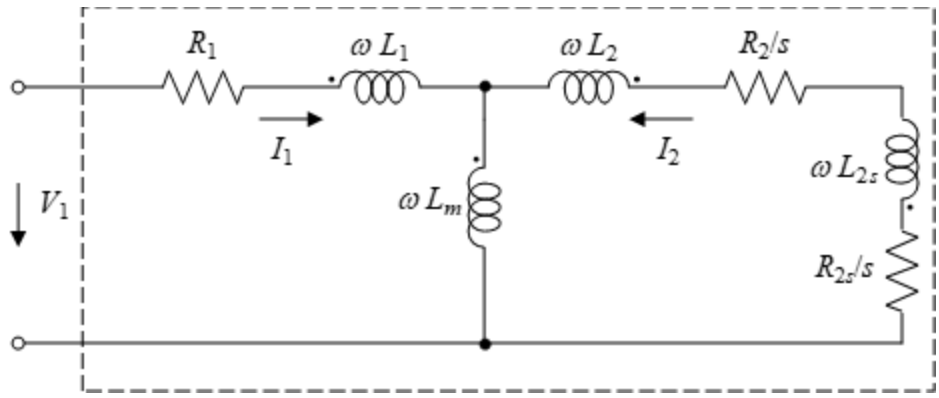
$$\xi = \xi_0\sqrt{s}$$

In the above equations, I_1 is the stator phase current and s is the rotor slip, L_{10}, L_{1m}, L_{20}, L_{2m}, I_{T1}, I_{T2}, R_{2s0}, L_{2s0} and ξ_0, as well as R_2 in the circuit, are 10 constant parameters identified from the eddy-current FEA solutions at different stator currents and rotor slips. If the stator or rotor leakage inductance is constant, L_{1m} or L_{2m} is zero, and I_{1T} or L_{T2} is not used in such a case.

The non-linear main inductance in the circuit is expressed as

$$L_m = L_{m0}/k_{sm}$$

where, L_{m0} is a constant parameter, and k_{sm} is the saturation factor of main field which is expressed as a look-up table. Both L_{m0} and k_{sm} are identified from the eddy current FEA solutions with different stator currents at no-load operation (with rotor slip being near zero).



Since R_{2s0} and L_{2s0} are slip dependent, the above circuit can be implemented in the frequency domain only. If the rotor bar is divided into two sections with top section ratio of k_1, and the bottom section ratio of $k_2 = 1 - k_1$, the resistance and self-inductance of the top section will be:

$$\begin{cases} R_{2t} = R_{2s0}/k_1 \\ L_{2t} = k_1 L_{2s0} \end{cases}$$

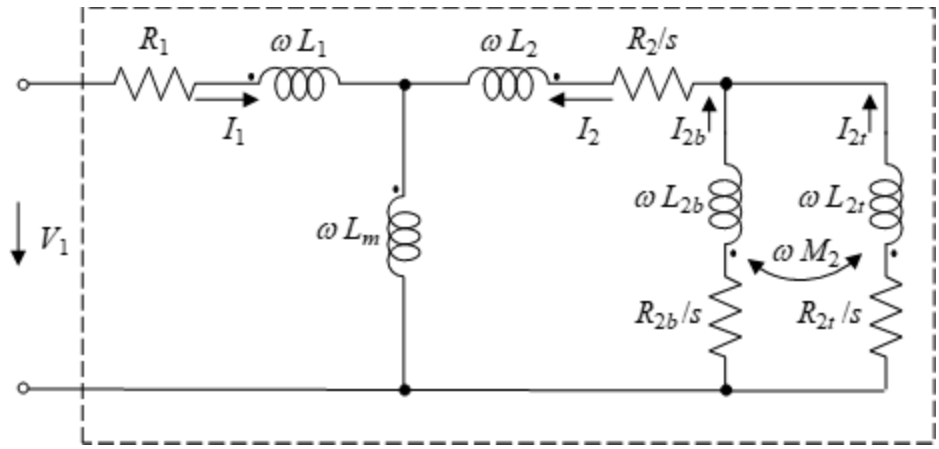
and the resistance and self-inductance of the bottom section are:

$$\begin{cases} R_{2b} = R_{2s0}/k_2 \\ L_{2b} = k_2 L_{2s0} + 3k_1 L_{2s0} \end{cases}$$

The mutual inductance between the top and bottom sections is:

$$M_2 = \frac{3}{2} k_1 L_{2s0}$$

The circuit with distributed rotor sections is shown in the following figure:



After all parameters in the first circuit are identified, the top section ratio k_1 is identified by minimizing the error of the impedance between the two circuits. All identified parameters are saved in file **ece_table.txt**, as shown in the figure below.

```
B_GeneralData
 Poles   4
 Phases 3
 Frequency        50
E_GeneralData

B_UnsatParam
 r1       4.27905
 l1e      0.00414949
 l10      0.00653594
 l1m      0
 it1      10
 r20      1.06698
 l20      0.0199702
 l2m      0.347714
 it2      0.310086
 r2s0     0.272791
 l2s0     0.00449169
 kxi0     2.78555
 k1       0.309297
 lm0      0.632328
 l00      0.0397237
 skew     0
 err      0.0319919
 use_tanh         0
E_UnsatParam

B_SatFactorForXm0_im_ksm 33
 0        1
 0.15625          1
 0.3125           1.00013
 ......
E_SatFactorForXm0_im_ksm
```

In above figure, r_1, $l1_e$, and *skew* are directly obtained from ECEIM_Model inputs Res, IndE, and SkewAng, respectively, 10 parameters, l10, l1m, it1, r20, l20, l2m, it2, r2s0, l2s0, kxi0 (ξ0), and lm0, as well as the look-up table between B_SatFactorForXm0_im_ksm and E_ SatFactorForXm0_im_ksm, are identified based on the first circuit, and k1 is identified based on the second circuit. Parameter l00 is the zero-sequence inductance. The err value represents the relative error of impedances between the FEA solutions and the parameter-derived results. If the value of use_tanh is 1, the saturation factors for the stator and rotor leakage inductances will be expressed as:

$$\begin{cases} k_{s1}\left(I_1\right) = \left(I_1/I_{T1}\right)/tanh(I_1/I_{T1}) \\ k_{s2}\left(I_1\right) = \left(I_1/I_{T2}\right)/tanh(I_1/I_{T1}) \end{cases}$$

to achieve smaller identification error, where tanh() is a hyperbolic tangent function.

Core Loss Parameters

Core losses can be computed for each sweep in eddy-current solver. To include core loss computation, users must check the **Core Loss** setting in the **Excitations/Set Core Loss** panel. Since core losses are computed in a post-process, including core loss computation will not affect the parameter identification of the induction machine circuit.

The stator and/or rotor core losses can be obtained from lamination steel or power ferrite material. For lamination steel material, the stator core loss can be computed from the phase circuit variables and expressed as:

$$P_{fe1} = k_{hs} {E_m}^2 / f + k_{cs} {E_m}^2 + k_{es} {E_m}^{1.5}$$

where k_{hs}, k_{cs}, and k_{es} are to-be-determined parameters, E_m is the phase induced voltage magnitude, and f is the frequency. For power ferrite material, core loss is expressed as:

$$P_{fe1} = C_m {E_m}^A f^B$$

where C_m, A, and B are to-be-determined parameters. The three parameters for either material can be identified at different induced voltage magnitudes and different frequencies by minimizing the error of core losses between the FEA solutions and the results obtained from the above equations. One of the above two equations will be automatically selected according to their identified errors. If the identified error based on the first equation is smaller than that based on the second equation, the first equation for lamination steel material will be selected. Otherwise, the second equation for power ferrite material will be selected. The core loss resistance is derived from core loss, and for lamination steel material, it is given by:

$$R_{fe1} = \frac{{E_m}^2/2}{P_{fe1}/3} = \frac{1.5}{k_{hs}/f + k_{cs} + k_{es}/{E_m}^{0.5}}$$

For power ferrite material, core loss resistance is given by:

$$R_{fe1} = \frac{{E_m}^2/2}{P_{fe1}/3} = \frac{1.5}{C_m {E_m}^{A-2} f^B}$$

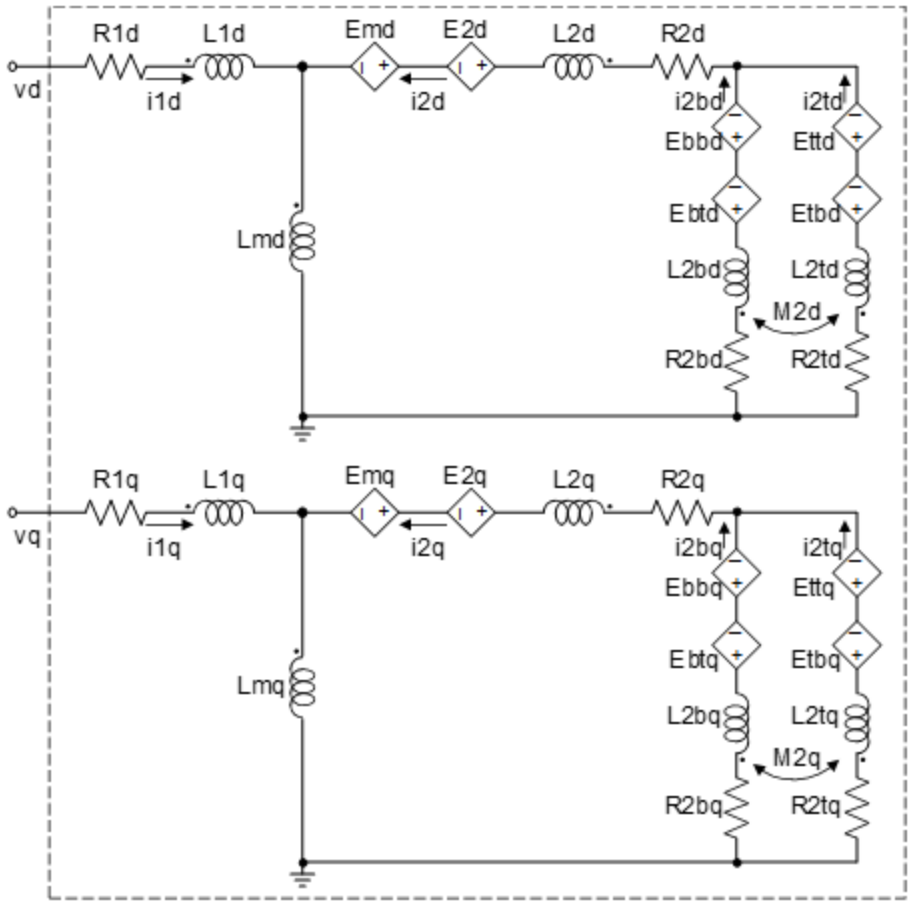
Replacing E_m by sE_m and f by sf, where s is the rotor slip, we can calculate rotor core loss in the same way as the stator core loss. Then the rotor core loss resistance for lamination steel material is:

$$R_{fe2} = \frac{{E_m}^2/2}{P_{fe2}/3} = \frac{1.5}{s k_{hs}/f + s^2 k_{cs} + s^{1.5} k_{es}/{E_m}^{0.5}}$$

and the rotor core loss resistance for power ferrite material is:

$$R_{fe2} = \frac{{E_m}^2/2}{P_{fe2}/3} = \frac{1.5}{C_m {E_m}^{A-2} f^B s^{A+B}}$$

Core loss resistance is frequency-dependent. In a time-domain circuit, as shown in the following figure, what we can measure are time-instant values. In order to use frequency-dependent resistance in a time-domain circuit, we need to calculate the frequency based on measurable time-instant values.



By measuring the induced voltages across Lmd and Lmq at time t, noted as Vmd and Vmq, respectively, we can compute the magnitude of induced voltage as:

$$V_m = \sqrt{V_{md}{}^2 + V_{md}{}^2}$$

Similarly, by measuring the flus linkages of Lmd and Lmq at time t, noted as λmd and λmq, respectively, we can compute the magnitude of flux linkage as:

$$\lambda_m = \sqrt{\lambda_{md}{}^2 + \lambda_{mq}{}^2}$$

The frequency then can be obtained from:

$$f = \frac{V_m}{2\pi\lambda_m}$$

and the induced voltage magnitude in three-phase side E_m for core loss resistance computation can be obtained by V_m with a factor of two-phase to three-phase transformation. The slip *s* for the computation of rotor core loss resistance can be obtained from the number of poles, rotor speed, and frequency.

The identified core loss parameters are listed between B_CorelossParam and E_CorelossParam in **ece_table.txt** file, as shown below.

```
B_CorelossParam
 typeS  3
 khs    0.00372061
 kcs    1.32565e-05
 kes    0.000768336
 errS   0.140547
 typeR  1
 khr    0.000180694
 kcr    5.1108e-06
 ker    -5.19829e-05
 errR   0.00768287
E_CorelossParam
```

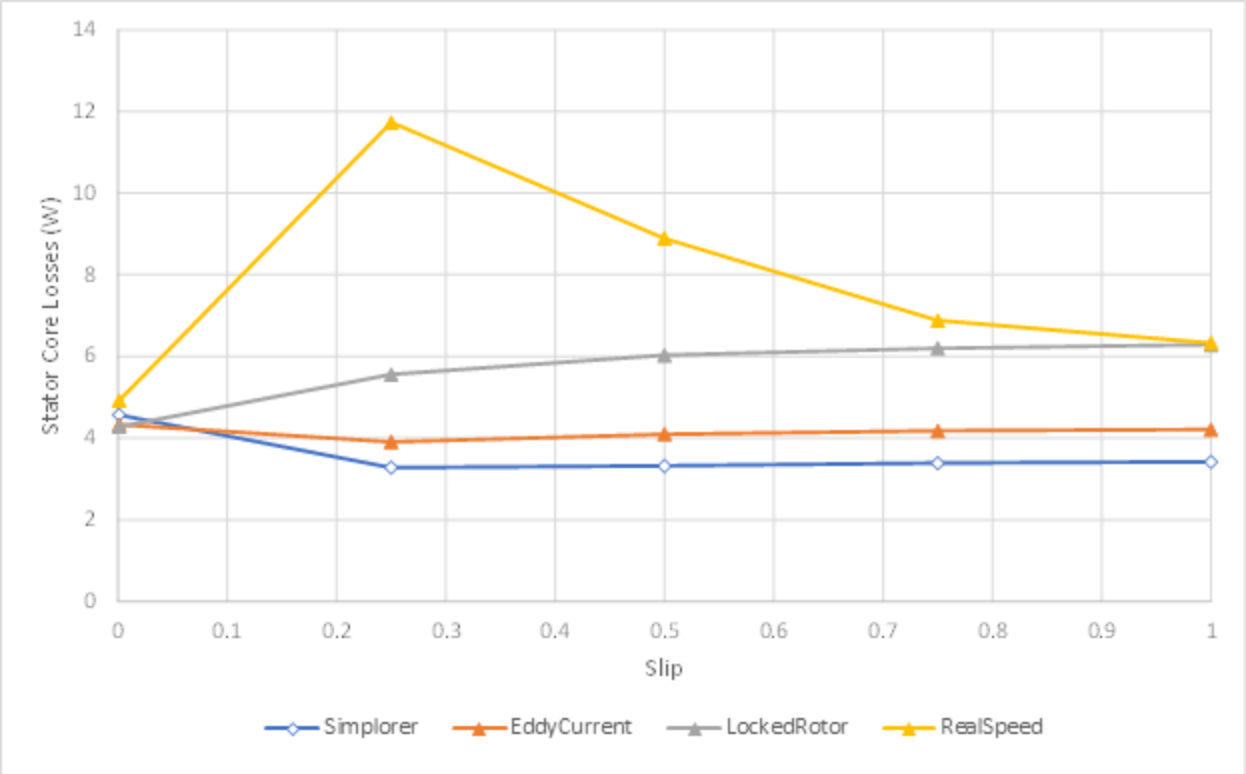
In the above figure, parameters typeS, khs, kcs, kes, and errS are for stator core loss. When typeS is 0, the circuit model will not include the stator core loss. When typeS is 1 or 3, the parameters are lamination-steel-based, and when it is 2 or 4, the parameters khs, kcs, and kes represent Cm, A, and B, respectively, for power ferrite material. When typeS is 1 or 2, two stator core loss resistances are connected in parallel with Lmd and Lmq, otherwise, when it is 3 or 4, two core loss resistances are connected from the input nodes of L1d and L1q to the ground. The errS value is the regression error for the stator core loss. Similarly, parameters typeR, khr, kcr, ker, and errR are for rotor core loss with the same definition as the parameters for stator core loss.

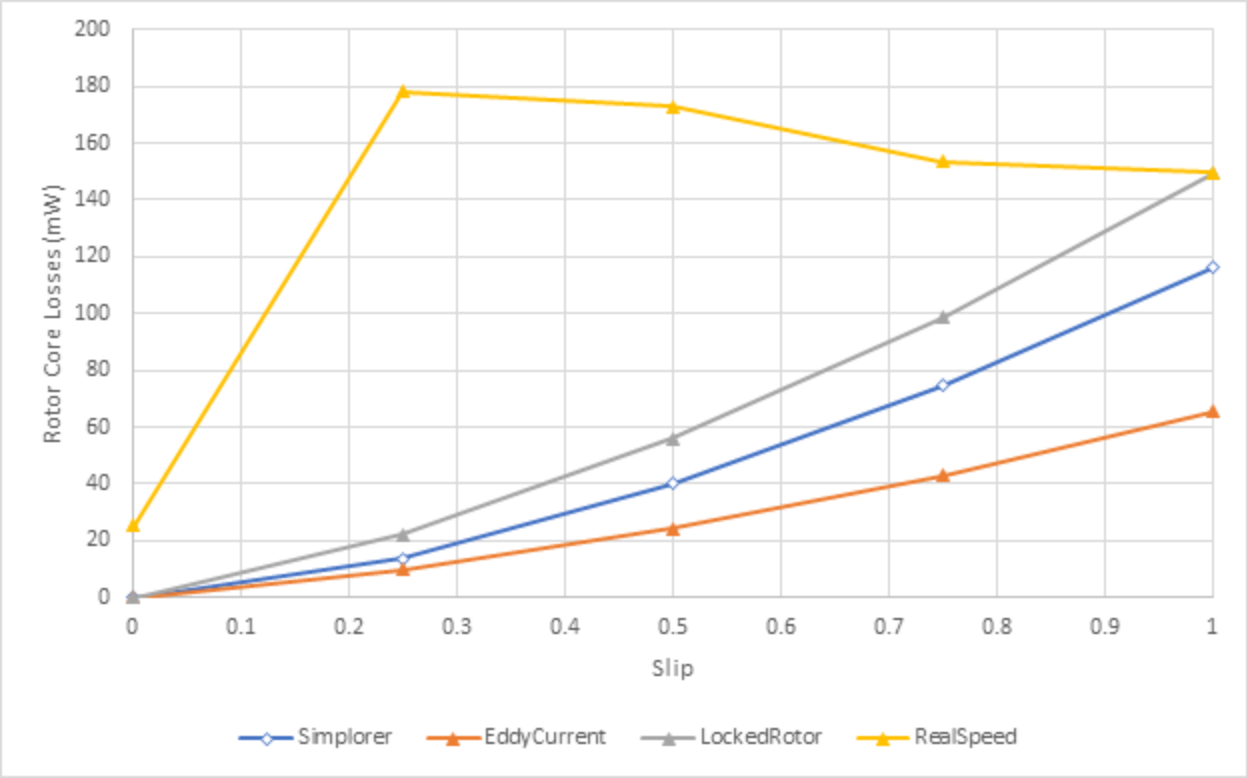
Since core loss parameters are identified from Maxwell eddy-current solutions at no-load operation based on the method referring the rotor frequency from the real operation condition to a locked-rotor operation (called locked-rotor method), the predicted core loss from the identified parameters will be under-estimated due to the following approximation errors:

1. The error caused by field difference between no-load and load operations. At no-load operation, the field in an induction machine is mainly the main flux which links both the stator and rotor windings, while at load operation, the main fluxes created by the stator and rotor currents will mostly offset each other, remaining large leakage fluxes which will go through the stator and rotor tooth tips. In the stator, the leakage flux will add to the main flux, causing the stator core loss increasing. In the rotor, the leakage flux will partially cancel the main flux, causing the rotor core loss decreasing a little.
2. The error caused by difference between sinusoidal (eddy-current solver) and non-sinusoidal (transient solver) waveforms. In eddy-current solver, core loss is computed based on pure sinusoidal waveforms, but in transient solver, core loss is computed based on real non-sinusoidal waveforms due to non-linearity. The harmonic field components will cause additional core losses in both stator and rotor cores in transient analysis.
3. The error caused by difference between locked-rotor and rotating operations. When rotor is rotating, the slot effects will cause additional core losses at the surfaces of the stator and rotor cores known as surface losses, as well as additional core loss at teeth of the stator and rotor cores known as pulsation losses.

The above three loss errors can be illustrated by the following two figures for stator and rotor core losses, respectively. In the figures, curves labeled with “Simplorer” are obtained from the system

simulation based on the extracted core loss parameters, curves labeled with “EddyCurrent” are obtained from the eddy-current simulation based on the locked-rotor method, curves labeled with “LockedRotor” are obtained from the transient simulation based on the locked-rotor method, and curves labeled with “RealSpeed” are obtained from the transient simulation with real rotor speed. The loss difference between curves “EddyCurrent” and “Simplorer” represents the error described in (1), the loss difference between curves “LockedRotor” and “EddyCurrent” indicates the error described in (2), and the loss difference between curves “RealSpeed” and “LockedRotor” is the error described in (3), which will increase as the stator currents increase, and will decrease as the rotor speed decreases.





Therefore, compare to Maxwell transient analysis, the core loss obtained from the circuit simulation will be under-estimated, especially at load operation. To adjust the simulated core loss, we introduce an argument KPfe in SIMPLORER (Twin Builder) circuit model to scale core losses for both stator and rotor. The default value of KPfe is 1.0. The reasonable value of KPfe at full-load operation may range from 1.5 to 2.5.

Since the stator and rotor core loss resistances are non-linear and frequency dependent, we allow users to consider or not consider core loss effects on circuit variables to avoid possible divergence during circuit simulation. When we input a positive value for KPfe, we will consider core loss effects by connecting the core loss resistances with derived values. When the input value for KPfe is negative, the core loss resistances are disconnected.

No matter core loss effects are considered or not, the stator and/or rotor core losses are output via circuit arguments Pfe1 and/or Pfe2 in the SIMPLORER (Twin Builder) circuit model.

ECEL_Model

The ECEL model is used to set up the sweeping of mover position of a translational motion. To change the name of the model placed on the sheet, click the symbol on the sheet, and change the name of the component in the property window (Value field in the DeviceName line, with the Parameter Values tab selected).

ECEL_Model1

The ECEL_Model contains the following parameters:

- DeviceName − the name of the device model with the default name as ECEL_Model1;
- MotionMax − the maximum motion range for sweeping;
- MotionIntervals − the number of uniform sweeping intervals for motion range;
- MotionType − the motion type: 0 for limited motion; 1: for periodic motion with full sweeping; 2 or more: for periodic motion with half or less sweeping.
- MotionDir − the motion direction in (x, y, z);

The initial position set in Maxwell designs is always referred to as 0 position in ECE model. When MotionType >= 1 (for periodic motions), the number of sweepings is equal to MotionIntervals. In such a case, the last point at MotionMax is not swept and is automatically extended from the first sweeping point at zero position based on the periodic conditions. When MotionType = 0 (for limited motion), the number of sweepings is equal to MotionIntervals + 1. In such a case, the motion limit is always ranged from 0 to MotionMax in the ECE model.

An ECEL_Model cannot be used individually. It should be used combined with an ECEW_Model and/or an ECE3_Model for rotational electrical machines.

For ECE model creation, the setup of Stop Time and Time step are ignored. At Maxwell run time, the circuit simulator assigns currents for all windings step-by-step based on the lists of current and position sweeps until all sweeps are finished. An ECE model is created automatically. You can import this ECE model in Twin Builder via the **Twin Builder>SubCircui>/Maxwell Component>Add Equivalent Circuit** menu command. For more information on coupling Maxwell designs to Twin Builder components, refer to the *Maxwell Equivalent Circuit Component* topic in the Twin Builder Help.

Note	You can add an ECEL_Model to an existing external circuit without deleting the original circuit. In such cases, the original circuit is ignored and can be kept for future use. If you want to recover the original circuit for a transient simulation, instead of ECE model creation, you just need to deactivate each ECE related model by right-clicking the model and selecting Deactivate in the short-cut menu.

For more information on coupling Maxwell designs to Twin Builder components, refer to "Coupling Maxwell to a Twin Builder Component " on page 17-84 .

ECER_Model

The ECER model enables you to create an ECE model from Maxwell Transient solutions. Compared with the procedure for creating ECE model from Maxwell Magnetostatic solutions, creating an ECE model directly from Maxwell Transient solutions takes advantage of winding and motion setups in Maxwell Transient.

The ECER model is used to set up the sweeping of rotor position. To change the name of the model placed on the sheet, click the symbol on the sheet, and change the name of the component in the property window (Value field in the DeviceName line, with the Parameter Values tab selected).

The ECER_Model contains the following parameters:

- DeviceName –the name of the device model with default name ECER_Model1.
- RotAngMax – specifies the maximum rotating angle (in mechanical degrees) for sweeping.
- InElecDeg – check to define RotAngMax in elec. degrees, uncheck in mech. degrees
- RotAngIntervals – specifies the number of uniform sweep intervals for rotating angle.
- SkewAng – specifies the skew angle (in mechanical degrees) of the stator or rotor core for 2D only.
- Poles – specifies the number of poles of the machine.
- Slots – specifies the number of stator slots to determine cogging torque period (available when Show Hidden is checked).
- Output – specifies how the look-up table outputs the winding flux linkages. 0 for flux linkages in ABC; 1 for flux linkages in DQ0; 2 for inductances in DQ and field flux linkage (available when Show Hidden is checked).

An ECER_Model cannot be used individually. It should be used combined with an ECEW_Model and/or an ECE3_Model for rotational electrical machines. When DQFluxOutput is checked, it must be combined with an ECE3_Model.

For ECE model creation, the setup of Stop Time and Time step are ignored. At Maxwell run time, the circuit simulator assigns currents for all windings step-by-step based on the lists of current sweeps until all sweeps are finished. An ECE model is created automatically. You can import this ECE model in Twin Builder via the **Twin Builder>SubCircuit>Maxwell Component>Add Equivalent Circuit** menu command. For more information on coupling Maxwell designs to Twin Builder components, refer to the *Maxwell Equivalent Circuit Component* topic in the Twin Builder Help.

Note	You can add an ECER_Model to an existing external circuit without deleting the original circuit. In such cases, the original circuit is ignored and can be kept for future use. If you want to recover the original circuit for a transient simulation, instead of ECE model creation, you just need to deactivate each ECE related model by right-clicking the model and selecting Deactivate in the short-cut menu.

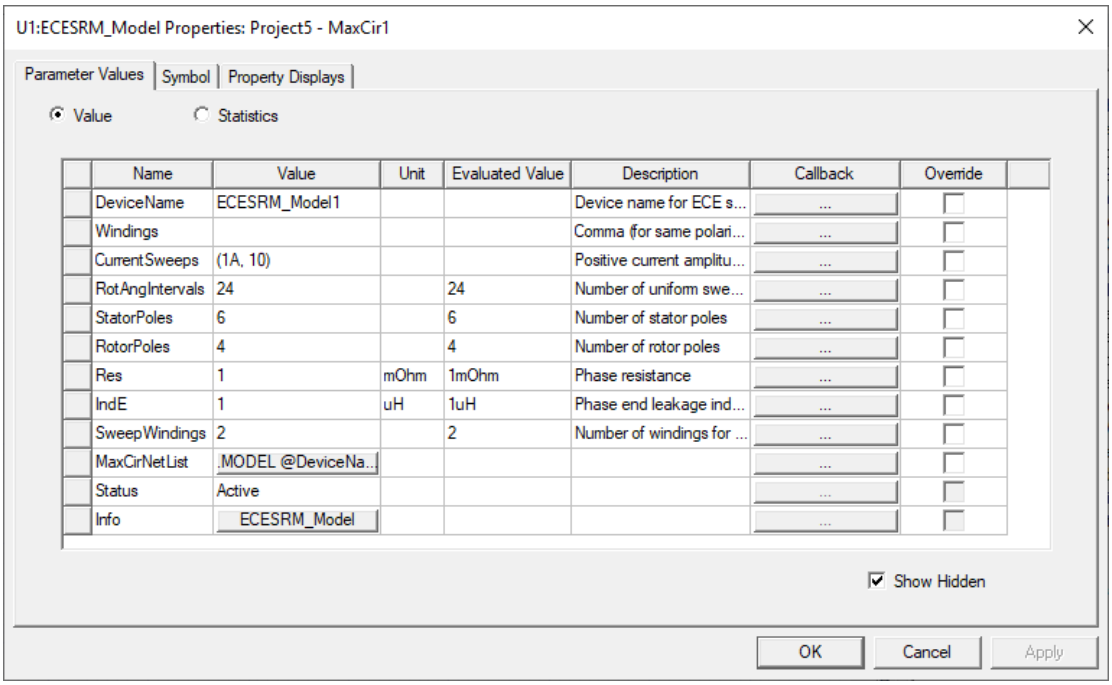
For more information on coupling Maxwell designs to Twin Builder components, refer to "Coupling Maxwell to a Twin Builder Component " on page 17-84 .

ECESRM_Model

The ECESRM model enables you to create an ECE model of a switched reluctance motor from Maxwell 2D and 3D Transient solutions.

To change the name of the model placed on the sheet, click the symbol on the sheet, and change the name of the component in the property window (Value field in the DeviceName line, with the Parameter Values tab selected).

ECESRM_Model1



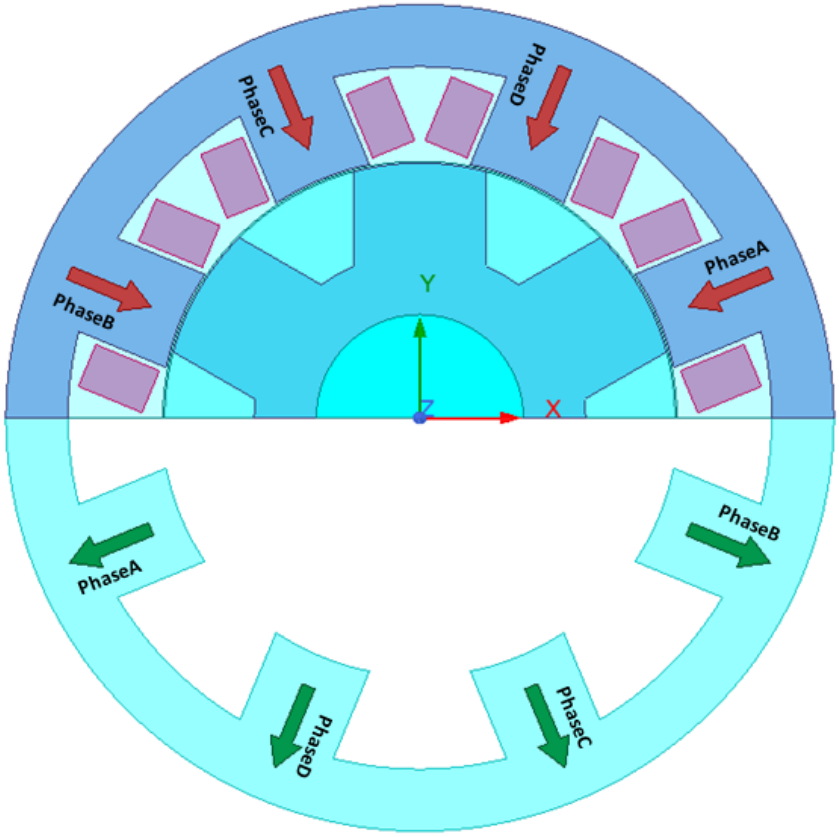
The ECESRM_Model contains the following parameters:

- DeviceName – the name of the device model with default name ECESRM_Model1.
- Windings – specifies winding names in the order of arrangement. Windings are separated by commas or semicolons. If two adjacent windings are defined with the same polarity, they will be separated by a comma “,”, otherwise by a semicolon “;”. For

example, for an 8/6 pole SRM, as shown below, windings for phase A, B, C and D are arranged in clockwise direction. The adjacent two windings of phase A and phase B have different polarities, and all other two adjacent windings have the same polarities. Therefore, windings can be specified in ECESRM_Model as:

“PhaseA; PhaseB, PhaseC, PhaseD”

where the separator between PhaseA and PhaseB is “;”, indicating different polarities between them. The last separator between PhaseD and PhaseA is not required because the total number of semicolons should be an odd number, and the last separator can be automatically determined.



- CurrentSweeps – specifies current sweeps in a winding. The first winding in the winding list is always excited with sweeping current. You just need to list 0 and all positive sweeping currents. All negative currents will be swept, or extended, symmetrically from the positive sweeping currents.
- RotAngIntervals – specifies the number of uniform sweeping intervals for rotor position in one rotor-pole region. For a 6-pole rotor, one rotor-pole region will be 60 mechanical degrees. If RotAngIntervals = 24, the sweeping step for rotor position will be 2.5 mechanical degrees.
- StatorPoles and RotorPoles – specifies the numbers of stator and rotor poles.
- Res and IndE – specifies winding resistance and end-leakage inductance for 2D.
- SweepWindings – specifies the number of windings to be excited. The value of SweepWindings could be 0, 1, or 2. Value 0 is for the case that only one winding current

is swept, and all mutual effects of other winding currents are ignored; value 1 is for the case that only one winding current is swept, and the mutual effects on winding flux linkage are considered, but the mutual effects of other windings currents on field saturation and torque are ignored; value 2 is for the case that two winding currents are swept, and all mutual effects are considered.

Look-up Table Processing

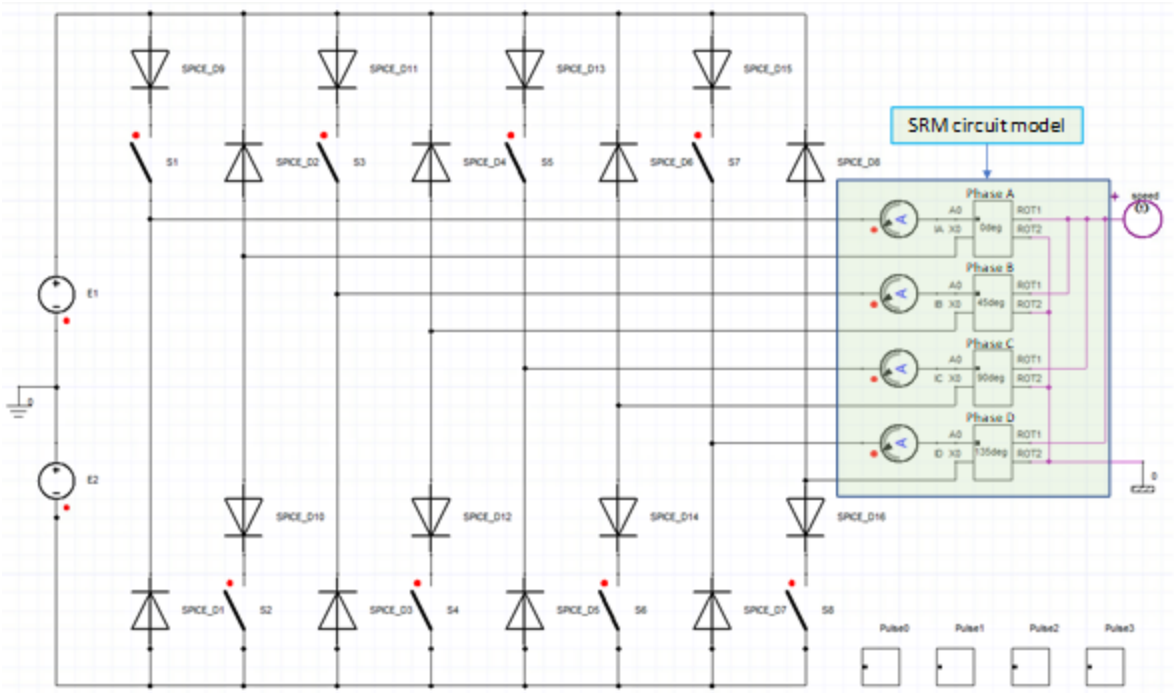
For an SRM, if a winding is triggered after the previously triggered winding has finished the free-wheeling process, there will be no mutual effects between the currently triggered winding and the previously triggered winding. In such a case, only one winding is excited with current at any time. In many cases, in order to increase power capacity, a winding could be triggered when the previous triggered winding is still working. In this situation, two windings could be excited with currents in some time periods. Therefore, we will provide different sweep types for different applications, which is specified by parameter SweepWindings, as described below.

- SweepWindings = 0

  SweepWindings = 0 means only one winding is excited, and all mutual effects between windings are ignored. The first listed winding PhaseA is excited with sweeping currents in the range of [0, Im], and the look-up table for the negative currents is automatically extended based on symmetry conditions. The final circuit model derived from this setting is equivalent to the circuit which is manually created by:

  i. Obtaining a single-phase circuit sub-model using ECEW_Model and ECER_Model.

  ii. connecting single-phase circuit sub-models for all phases with suitable IniPos settings, as shown in the circuit inside the shaded rectangle below.



This sweep type has the smallest look-up table, which includes controlling variables of one winding current and rotor position, and controlled variables of flux linkage and torque. For this sweep type, it is not necessary to solve the field for negative sweeping currents. The look-up table for the negative currents can be automatically extended based on symmetry conditions, as given below:

$$\begin{cases} F_{this}\left(-I_{this},\theta\right) = -F_{this}\left(I_{this},\theta\right) \\ T\left(-I_{this},\theta\right) = T\left(I_{this},\theta\right) \end{cases}$$

where $F_{this}(I_{this}, \theta)$ and $T(I_{this}, \theta)$, as outputs of look-up table, are the flux linkage of the excited winding and rotor torque, respectively, and the controlling variables I_{this} and θ are the winding current and rotor position, respectively.

- SweepWindings = 1

  SweepWindings = 1 means only one winding is excited, and the mutual effects on winding flux linkage are considered, but the mutual effects on field saturation and torque are ignored. In order to easily illustrate windings for look-up table creation, we call the excited winding, PhaseA, as *this* winding, PhaseB as *next* winding, and PhaseD as *pre*vious winding. In this way, in the first single-phase sub-model as shown in the figure above, *this*/*next*/*pre*vious windings represent phase A/B/D windings. In the second single-phase sub-model, they represent phase B/C/A windings, and so on. Only *this* winding has a circuit inside a single-phase sub-model. The mutual effects of the *next*/*pre*vious currents on this-winding flux linkage are considered by:

$$F = F_{this}\left(I_{this},\theta\right) \pm I_{next}\cdot M_{next}\left(I_{this},\theta\right) \pm I_{pre}\cdot M_{pre}\left(I_{this},\theta\right)$$

  where $F_{this}(I_{this}, \theta)$, $M_{next}(I_{this}, \theta)$, $M_{pre}(I_{this}, \theta)$, as outputs of look-up table, are flux linkage, mutual inductance between *this* and *next* windings, and mutual inductance between *this* and *pre*vious windings, respectively. Controlling variables I_{this} and θ represent *this*-winding current and rotor position, respectively. The *next* and *pre*vious winding currents, $I_{next}$and I_{pre}, respectively, are delivered via parameters of single-phase sub-models. The signs before the *next* and *pre*vious winding currents will be automatically determined in the final SRM model based on the separators for *pre*vious, *this* and *next* windings. Like the case of SweepWindings = 0, the look-up table for the negative currents can be automatically extended based on symmetry conditions, as given below:

$$\begin{cases} F_{this}\left(-I_{this},\theta\right) = -F_{this}\left(I_{this},\theta\right) \\ M_{next}\left(-I_{this},\theta\right) = M_{next}\left(I_{this},\theta\right) \\ M_{pre}\left(-I_{this},\theta\right) = M_{pre}\left(I_{this},\theta\right) \\ T\left(-I_{this},\theta\right) = T\left(I_{this},\theta\right) \end{cases}$$

  Comparing this with the sweep type of SweepWindings = 0, this sweep type has the same controlling variables with the same sweeping points, but it has a larger-size look-up table because it has more controlled variables.

- SweepWindings = 2

  SweepWindings = 2 means two windings, *this* and *next* windings, are excited, therefore all mutual effects on winding flux linkage, field saturation and torque are considered. If

CurrentSweep in the ECESRM_Model is specified in the range of [0, Im], the *this*-winding current will be swept in the range of [-Im, Im], and the *next*-winding current will be swept in the range of [0, Im]. After field simulations for all sweeps are completed, we obtain a clean look-up table. This look-up table has two controlled variables (*this*-winding flux linkage F_{this} and torque T); and three controlling variables (*this*-winding current I_{this}, *next*-winding current I_{next}, and rotor position θ). In this clean look-up table, T is the torque produced by *this*-winding current but considered the mutual effects of *next*-winding current. In field analysis, a torque can be derived based on field solutions and be expressed, by numerical integration over all related mesh elements, as:

$$T_m = \sum_{i=1}^{n} k_{Ti} \left(\mathbf{B}_i \cdot \mathbf{B}_i \right)$$

where $\mathbf{B_i}$ is the flux-density vector created by all winding currents, and k_{Ti} is a coefficient, in mesh element *i*. If we freeze the permeabilities of all non-linear materials, the field becomes linear. Then, the flux density vector in any mesh elements can be expressed as:

$$\mathbf{B}_i = \mathbf{B}_{1i} + \mathbf{B}_{2i}$$

where $\mathbf{B_{1i}}$ and $\mathbf{B_{2i}}$ are the flux-density vectors created by this and next winding currents, respectively. Hence, the torque will be:

$$T_m = \sum_{i=1}^{n} k_{Ti} \left(\mathbf{B}_{1i} + \mathbf{B}_{2i} \right) \cdot \left(\mathbf{B}_{1i} + \mathbf{B}_{2i} \right) = T_{11} + 2T_{12} + T_{22}$$

with

$$\begin{cases} T_{11} = \sum_{i=1}^{n} k_{Ti} \left(\mathbf{B}_{1i} \cdot \mathbf{B}_{1i} \right) \\ T_{12} = \sum_{i=1}^{n} k_{Ti} \left(\mathbf{B}_{1i} \cdot \mathbf{B}_{2i} \right) \\ T_{22} = \sum_{i=1}^{n} k_{Ti} \left(\mathbf{B}_{2i} \cdot \mathbf{B}_{2i} \right) \end{cases}$$

In field analysis, we can get T_{11} by injecting current in this winding only, and we get T_{22} by infecting current in next winding only. Then we get:

$$T_{12} = (T_m - T_{11} - T_{22})/2$$

Finally, we get torque in the clean look-up table as:

$$T = T_{11} + T_{12}$$

The clean look-up table can be extended to a full look-up table based on even and odd symmetry conditions, as given below:

$$\begin{cases} F_{this} \left(I_{this}, -I_{next}, \theta \right) = -F_{this} \left(-I_{this}, I_{next}, \theta \right) \\ T \left(I_{this}, -I_{next}, \theta \right) = T \left(-I_{this}, I_{next}, \theta \right) \end{cases}$$

The flux linkage and torque in the look-up table are derived based on this-winding current with considering the mutual effects of next-winding current. In real operation with two excited windings, we need to get the flux linkage and torque produced by each winding

current. For example, in the circuit diagram above, if phases A and B are excited simultaneously, the flux linkage and torque, produced by phase-A current with mutual effects of phase-B current, can be directly obtained from the look-up table because phase-A current is *this*-winding current and phase-B current is *next*-winding current, as given below:

$$\begin{cases} F_1 = F_{this}\left(I_{this}, \pm I_{next}, \theta\right) \\ T_1 = T\left(I_{this}, \pm I_{next}, \theta\right) \end{cases}$$

where the sign before I_{next} is positive if the separator between *this* and *next* windings is the same as that between the first and second windings listed in the Windings field in the properties dialog box above, or negative otherwise. Since the look-up table includes the mutual-effect information, it is important to transfer the *next*-winding current to make sure the polarity relationship between *this* and *next* windings in a single-phase sub-model is the same as that for look-up table creation. For example, for the phase-A sub-model in the circuit diagram above, this sign is positive because the separator between *this* and *next* windings, or between PhaseA and PhaseB, is ";", which is the same as that between PhaseA and PhaseB for look-up table creation. However, for the phase-B sub-model, the sign is negative because the separator between *this* and *next* windings, or between PhaseB and PhaseC, is ",", which is different from the separator ";" between PhaseA and PhaseB for look-up table creation.

When we consider the flux linkage and torque produced by phase-B current with mutual effects of phase-A current, we cannot obtain them directly from the look-up table because in this situation phase-B current is *this*-wingding current and phase-A current is *pre*-winding current, but we can still obtain them from the look-up table based on symmetry conditions. In the winding polarities diagram above, if we mirror the geometry about the center line of the phase-B pole, then phase-A winding aligns with phase-C winding, and rotor position becomes negative. Based on this symmetry condition, we have:

$$\begin{cases} F_2 = F_{this}\left(I_{this}, \pm I_{pre}, -\theta\right) \\ T_2 = -T\left(I_{this}, \pm I_{pre}, -\theta\right) \end{cases}$$

where the sign before I_{pre} is positive if the separator between *this* and previous windings is the same as that between the first and second windings listed in the Windings field in the properties dialog box above, or negative otherwise.

In the final SRM model, the signs before *Inext* and Ipre for all single-phase sub-models will be automatically determined based on all separators listed in the Windings field, which, together with *Inext* and *Ipre*, are forwarded to sub-models via input parameters.

You can import this ECE model in Twin Builder via the **Twin Builder>SubCircuit>Maxwell Component>Add Equivalent Circuit** menu command. For more information on coupling Maxwell designs to Twin Builder components, refer to the *Maxwell Equivalent Circuit Component* topic in the Twin Builder Help.

Note	You can add an ECESRM_Model to an existing external circuit without deleting the original circuit. In such cases, the original circuit is ignored and can be kept for future use. If you want to recover the original circuit for a simulation, instead of ECE model creation, you just need to deactivate each ECE related model by right-clicking the

	model and selecting Deactivate in the short-cut menu.

For more information on coupling Maxwell designs to Twin Builder components, refer to the *Coupling Maxwell to a Twin Builder Component* topic in the Maxwell Help

ECET_Model

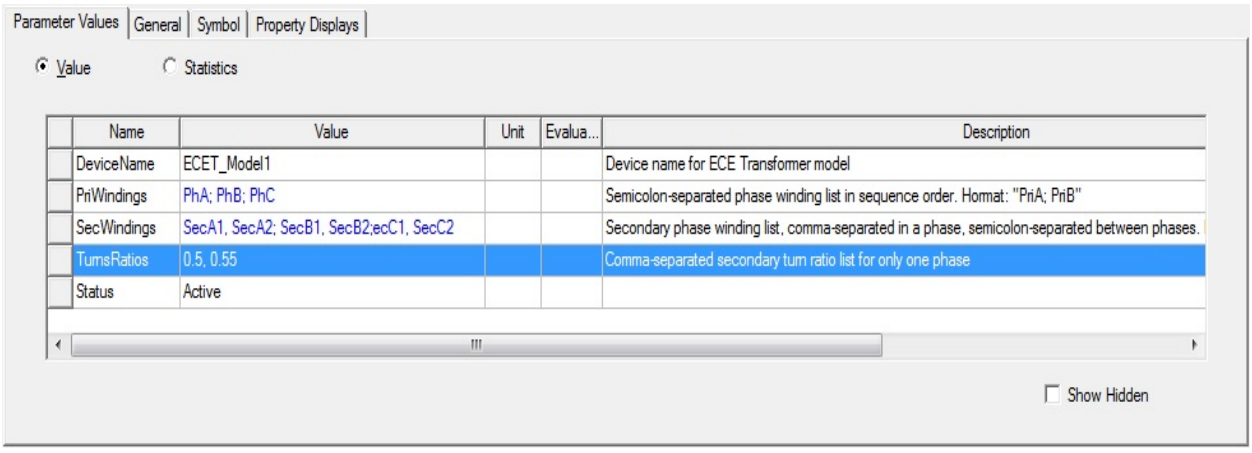
The ECET model is used to set up the sweeping of the magnetizing current for each phase of a single- or poly-phase transformer. To change the name of the model placed on the sheet, click the symbol on the sheet, and change the name of the component in the property window (Value field in the DeviceName line, with the Parameter Values tab selected).

ECET_Model1

The ECET_Model contains the following parameters:

- DeviceName − the name of the device model with the default name as ECET_Model1;
- PriWindings − the name list of all primary windings with semicolons for separators between phase;
- SecWindings − the name list of all secondary windings with semicolons for separators between phases;
- TurnsRatios − the ratio of the secondary turns to the primary turns for only one phase.

If a transformer has two or more secondary windings for one phase, the secondary winding name list is separated by commas between winding names for the same phase, and TurnsRatios is a value list, instead of a single value, which is separated by commas between values for one phase. For example, for a three-phase transformer with two secondary windings for each phase, the parameters for ECET_Model are set as:



An ECET_Model cannot be used individually. It should be used combined with an ECEW_Model and/or an ECE3_Model.

In an ECE model look-up table, only the magnetizing inductance for each phase is assumed to be non-linear, and all leakage inductances for all primary and secondary windings are assumed to be constant. Therefore, only the magnetizing current of the primary winding for each phase is required to be swept, which reduces the number of sweeping points significantly.

For ECE model creation, the setup of Stop Time and Time step are ignored. At Maxwell run time, the circuit simulator assigns currents for all windings step-by-step based on the list of current sweeps until all sweeps are finished. An ECE model is created automatically.

You can import this ECE model in Twin Builder via the **Twin Builder>SubCircuit>Maxwell Component>Add Equivalent Circuit** menu command. When the ECET_model is created and then imported into Twin Builder the following additional parameters are made available for use in Twin Builder:

- **r1*_[WindingName]*** is winding resistance (default = 0)
- **l1*_[WindingName]*** is end leakage inductance (default = 1mOhm, for 3D always = 0)
- **i0*_[WindingName]*** is initial current (default = 0)

For more information on coupling Maxwell equivalent circuit designs to Twin Builder components, refer to the *Maxwell Equivalent Circuit Component* topic in the Twin Builder Help.

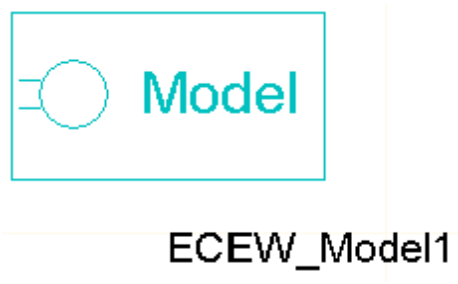
Note	You can add an ECET_Model to an existing external circuit without deleting the original circuit. In such cases, the original circuit is ignored and can be kept for future use. If you want to recover the original circuit for a transient simulation, instead of ECE model creation, you just need to deactivate each ECE related model by right-clicking the model and selecting Deactivate in the short-cut menu.

For more information on coupling Maxwell designs to Twin Builder components, refer to "Coupling Maxwell to a Twin Builder Component " on page 17-84 .

ECEW_Model

The ECEW model enables you to create an ECE model from Maxwell Transient solutions. Compared with the procedure for creating ECE model from Maxwell Magnetostatic solutions, creating an ECE model directly from Maxwell Transient solutions takes advantage of winding and motion setups in Maxwell Transient.

The ECEW model is used to set up the sweeping of one winding current. To change the name of the model placed on the sheet, click the symbol on the sheet, and change the name of the component in the property window (Value field in the DeviceName line, with the Parameter Values tab selected).



The ECEW_Model contains the following parameters:

- DeviceName – the name of the device model with default name ECEW_Model1.
- WindingName – specifies the name of winding whose current is to be swept.
- CurrentSweeps – specifies lists of all sweep currents. The format is: “v1, v2, (dv1, n1), v3, …”, where v1, v2, v3 specify individual sweep values, and (dv1, n1) specify a sweeping range with dv1 being the increment, and n1 the number of increments. For example, a list of “-2.5A, (0.5A, 10)” is equivalent to "-2.5A, -2A, -1.5A, -1A, -0.5A, 0, 0.5A, 1A, 1.5A, 2A, 2.5A", and a list of "(0.5A, 5)" is equivalent to “0, 0.5A, 1A, 1.5A, 2A, 2.5A”.
- Res – phase winding resistance (available when Show Hidden is checked).
- IndE – phase end-leakage inductance (available when Show Hidden is checked).

An ECEW_Model can be used individually to create the ECE model of a single-phase inductor, or combined with other ECEW_Models to setup two or more windings for poly-phase inductors. It can also be combined with an ECER_Model, as well as an ECE3_Model, for rotational electrical machines.

For ECE model creation, the setup of Stop Time and Time step are ignored. At Maxwell run time, the circuit simulator assigns currents for all windings step-by-step based on the lists of current sweeps until all sweeps are finished. Then an ECE model is created automatically. You can import this ECE model in Twin Builder via the **Twin Builder>SubCircuit>Maxwell Component>Add Equivalent Circuit** menu command. For more information on coupling Maxwell designs to Twin Builder components, refer to the *Maxwell Equivalent Circuit Component* topic in the Twin Builder Help.

Note	You can add an ECEW_Model to an existing external circuit without deleting the original circuit. In such cases, the original circuit is ignored and can be kept for future use. If you want to recover the original circuit for a transient simulation, instead of ECE model creation, you just need to deactivate each ECE related model by right-clicking the model and selecting Deactivate in the short-cut menu.

For more information on coupling Maxwell designs to Twin Builder components, refer to "Coupling Maxwell to a Twin Builder Component " on page 17-84 .

Passive Elements

The following passive elements are available in the Maxwell Circuit Elements component library:

- Cap: Capacitor
- DIODE: Diode
- DIODE_Model: Diode Model Data
- Ind: Inductor
- IndM: Mutual Inductance
- Res: Resistor
- SW_I: Current Controlled Switch
- SW_I4: Current Controlled Switch with Controlling Port
- SW_IModel: Model Data for Current Controlled Switches
- SW_V: Voltage Controlled Switch
- SW_V4: Voltage Controlled Switch with Controlling Port
- SW_VModel: Model Data for Voltage Controlled Switches
- Transformer: Ideal Transformer

The text before the colon (**:**) represents the component name and can be changed in the **Properties** window once the component is placed in the schematic.

Related Topics

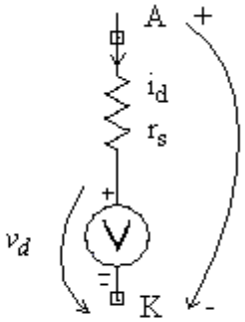
Assigning Component Properties in Maxwell Circuit Editor

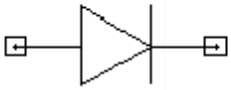
Diode & Diode Model

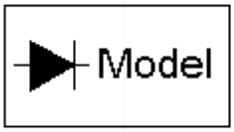
Diode element must always be used together with a diode model. One diode model element can be used as reference for multiple diodes, as needed. Thus, once a diode is placed on the design sheet, a corresponding model must also be present on the sheet. Once both needed elements (diode and diode model) have been placed on the sheet, right mouse click the diode model and specify the parameters as required by the application. Then, right mouse click the diode and create the reference to the corresponding model by entering the name of the model in the MOD line (Parameter Values tab should be selected). If you select the **Show Hidden** check box, the AREA diode parameter (used below in the model definition) becomes visible. The default value of the AREA parameter is 1.

The diode model used by Maxwell Circuit Editor is a static model as described by the following equation:

$$i_d = i_s \cdot \left(e^{v_d / v_t} - 1 \right) - \left(i_{bv} \cdot e^{-(v_d + BV)v_t} \right)$$







$$v_t = N \cdot (kT)/q$$

$$q = 1.6022 \cdot 10^{-19}\, C$$

$$k = 1.3807 \cdot 10^{-23}\, (J/K)$$

T is temperature in K, fixed at 300 K

$$i_s = IS \cdot AREA \cdot e^{\left(\frac{T/T_0 - 1}{kT - 1}\right) \cdot (EG)/v_t} \cdot \left(\frac{T}{T_0}\right)^{\frac{XTI}{N}}$$

$$i_{bv} = IBV \cdot AREA$$

$$r_s = \frac{RS}{AREA}$$

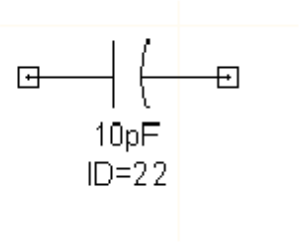
The model parameters are as follows:

- *IS* is the saturation current in Amps.
- *RS* is contact resistance in Ohms.
- *N* is the emission coefficient.

- *EG* is the barrier height at 0 K, in volts.
- *XTI* is the diode saturation current temperature coefficient.
- *BV* is the magnitude of the reverse breakdown voltage in volts.
- *IBV* is the magnitude of the reverse breakdown current in amps.
- *TNOM* is the reference temperature in Celsius.

Capacitor (CAP)

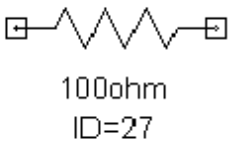
A capacitor is assumed to be ideal (without losses or inductance) and is defined by the value of its capacitance (in the unit chosen by the user) and the corresponding initial condition (initial voltage). The straight bar of the capacitor symbol is used as the positive reference, and the curved bar of the capacitor symbol is used as a negative reference for the initial voltage (expressed in volts). The default value of the initial voltage for all capacitors is zero. By default, the capacitance and element ID are displayed next to the component.



Resistor (Res)

Resistor is assumed to be ideal (without inductive or capacitive effects) and is defined by the value of its resistance (in the unit chosen by the user, Ohm by default.

By default the resistance and element ID are displayed next to the component.

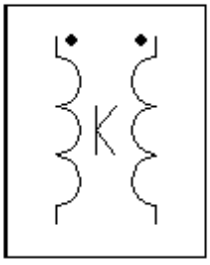


Inductor (IND)

Inductor is assumed to be ideal (without resistive or capacitive effects) and is defined by the value of its inductance (in the unit chosen by the user) and the corresponding initial condition (initial current). Note that the dot next to the inductor symbol is used as the positive reference for the initial current (positive current is oriented from the "dotted" terminal toward to "un-dotted" terminal of the inductor, through the inductor). The dot is also used to specify the mutual inductance between two or more inductors and thus determines the "polarized" terminals of the inductors. The default value of the initial current for all inductors is zero. By default the inductance and element ID are displayed next to the component.

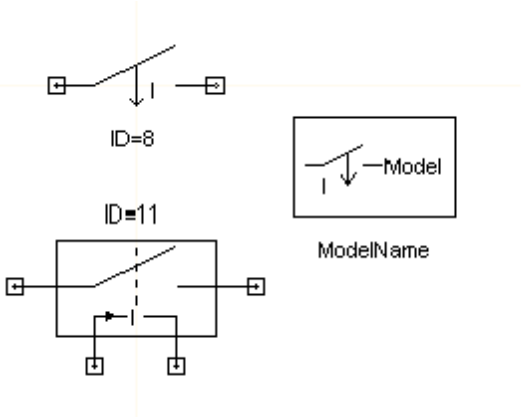
Mutual Inductance (IndM)

Mutual inductance is used to specify the inductive coupling between inductors. It is defined by specifying the IDs of the coupled inductors (always in pairs) and the coupling coefficient, a number between -1 and 1. The default value of the coupling coefficient is 0.95.



Current Controlled Switch (SW_I)

The current controlled switch comes in two flavors: with controlling port and without a controlling port. In either case a model data for the current controlled switch needs to be specified, similarly to the case of the diode.



Right-click the current controlled switch model and select Properties. With the Parameter Values tab selected, specify the switch model name (in the MOD line) as well as the ID of the controlling element: either an ammeter or a voltage source. In the later case the controlling quantity is the current through the voltage source. (Note that an ammeter is a voltage source with zero voltage, i.e. a short circuit).

The current controlled switch with controlling port allows for the controlling quantity to be wired directly using connections with wires. In this case a reference arrow in the controlling port is displayed and is internally used as current reference (positive current flow as indicated by the arrow).

In the model for the current controlled switch the following parameters are used:

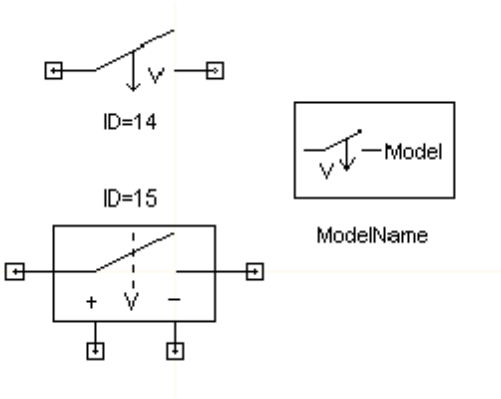
- *Ron* is the resistance of the switch in the **on** state (0.001 ohms default value).
- *Roff* is the resistance of the switch in the **off** state (1,000,000 ohms default value).
- *Ion* is the "**on**" value of the controlling current in amps. **If I>Ion, then R=Ron.**
- *Ioff* is the "**off**" value of the controlling current in amps. **If I<Ioff, then R=Roff.**

Note	Setting $\mathbf{R_{OFF} = 0}$ in the voltage or current controlled switch model data changes the behavior of the device into a **controlled conductance**, according to the following equation:

	$$G(P) = \frac{1}{R_{ON}} \cdot f_{control}(P)$$ where: $f_{control}(P)$ is a function describing the controlling signal -- a time, position, or speed dependent current source or voltage source. Function ***G(P)*** clearly shows that the magnitude of the conductance is dictated by both the value of R_{ON} and the magnitude of the control signal, while the time / position / speed dependency is dictated by the control signal itself.
Note	Setting $R_{ON} = 0$ in the voltage or current controlled switch model data changes the behavior of the device into a **controlled resistance**, according to the following equation: $R(P) = R_{OFF} \cdot f_{control}(P)$ where: $f_{control}(P)$ is a function describing the controlling signal -- a time, position, or speed dependent current source or voltage source. Function ***R(P)*** clearly shows that the magnitude of the conductance is dictated by both the value of R_{OFF} and the magnitude of the control signal, while the time / position / speed dependency is dictated by the control signal itself.

Voltage Controlled Switch (SW_V)

The voltage controlled switch comes in two flavors: with controlling port and without a controlling port. In either case a model data for the voltage controlled switch needs to be specified, similarly to the case of the diode and current controlled switches.



Right-click the voltage controlled switch model and select Properties. With the Parameter Values tab selected, specify the switch model name (in the MOD line) as well as the ID of the controlling element: either a voltmeter or a current source. In the later case the controlling quantity is the voltage across the current source. (Note that a voltmeter is a current source with zero current, i.e. an open circuit).

The voltage controlled switch with controlling port allows for the controlling quantity to be wired directly using connections with wires. In this case the reference for the voltage across the controlling port is displayed by "+" and "-" symbols and the two symbols are internally used as voltage reference.

In the model for the voltage controlled switch the following parameters are used:

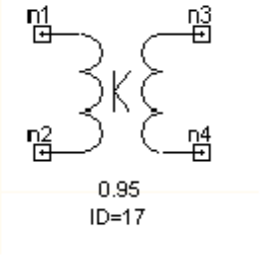
- Ron is the resistance of the switch in the **on** state (0.001 ohms default value).
- Roff is the resistance of the switch in the **off** state (1,000,000 ohms default value).
- Von is the "**on**" value of the controlling voltage in volts. **If V>Von, then R=Ron.**
- Voff is the "**off**" value of the controlling voltage in volts. **If V<Voff, then R=Roff.**

Note	Setting $\mathbf{R_{OFF} = 0}$ in the voltage or current controlled switch model data changes the behavior of the device into a **controlled conductance**, according to the following equation: $$G(P) = \frac{1}{R_{ON}} \cdot f_{control}(P)$$ where: $f_{control}(P)$ is a function describing the controlling signal -- a time, position, or speed dependent current source or voltage source. Function ***G(P)*** clearly shows that

	the magnitude of the conductance is dictated by both the value of R_{ON} and the magnitude of the control signal, while the time / position / speed dependency is dictated by the control signal itself.
Note	Setting $R_{ON} = 0$ in the voltage or current controlled switch model data changes the behavior of the device into a **controlled resistance**, according to the following equation: $R(P) = R_{OFF} \cdot f_{control}(P)$ where**:** $f_{control}(P)$ is a function describing the controlling signal -- a time, position, or speed dependent current source or voltage source. Function ***R(P)*** clearly shows that the magnitude of the conductance is dictated by both the value of R_{OFF} and the magnitude of the control signal, while the time / position / speed dependency is dictated by the control signal itself.

Transformer

The transformer is an ideal element of infinite power without resistive or capacitive effects and linear. It is defined the values of the primary and secondary inductances and the coupling coefficient between the two windings.



Probes

Three types of probes are available in the Maxwell Circuit Elements component library:

- Ammeter: Ammeter
- Voltmeter: Voltmeter
- VoltmeterG: Voltmeter with One Pin Grounded

The text before the colon (**:**) represents the component name and can be changed in the **Properties** window once the component is placed in the schematic.

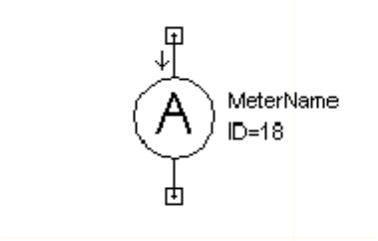
Related Topics

Assigning Component Properties in Maxwell Circuit Editor

Ammeter

The ammeter is an ideal element (equivalent with an ideal voltage source with zero voltage). An arrow is attached to the symbol so that a positive current measured by the ammeter flows as indicated by the arrow.

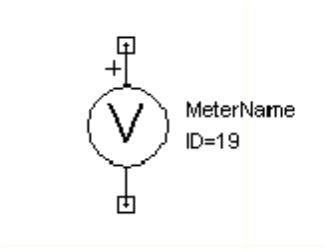
No numerical value is needed, just the element ID and name may by altered by the user.



Voltmeter

The voltmeter is an ideal element with two pins (equivalent with an ideal current source with zero current). A plus sign is attached to the voltmeter symbol so that a reference for the voltage measured by the voltmeter is possible.

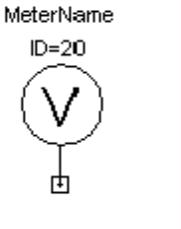
No numerical value is needed, just the element ID and name may by altered by the user.



Voltmeter with One Pin Grounded (VoltmeterG)

The voltmeter with one pin grounded is an ideal element with one pin, equivalent with an ideal current source with zero current. A plus sign is not necessary since it is assumed that the grounded pin (not available for connection in the circuit) is the negative one while the pin available for connection is the positive one. Thus, the reference for the voltage measured by the voltmeter is possible.

No numerical value is needed, just the element ID and name may by altered by the user.



Current and Voltage Sources

Sources available in the Maxwell Circuit Editor can be defined such that in general the dependency of the current or voltage can be made function of time, position, or speed. The type of dependency is part of the properties of the respective source and is always user-selectable. Thus, in the equation defining the behavior of the source (when specified), the variable "t" can mean TIME or POSITION or SPEED as selected by the user for each application.

The default dependency type is TIME.

The following types of sources are available in the Maxwell Circuit Elements component library:

- IDC: DC Current Source
- IExp: Exponential Current Source
- IPulse: Pulse Current Source
- IPWL: Piecewise Linear Current Source
- IPWM: PWM Current Source
- ISffm: Frequency Modulated Sinusoidal Current Source
- ISin: Sinusoidal Current Source
- VDC: DC Voltage Source
- VExp: Exponential Voltage Source
- VPulse: Pulse Voltage Source
- VPWL: Piecewise Linear Voltage Source
- VPWM: PWM Voltage Source
- VSffm: Frequency Modulated Sinusoidal Voltage Source
- VSin: Sinusoidal Voltage Source

The text before the colon (**:**) represents the component name and can be changed in the **Properties** window once the component is placed in the schematic.

Related Topics

Assigning Component Properties in Maxwell Circuit Editor

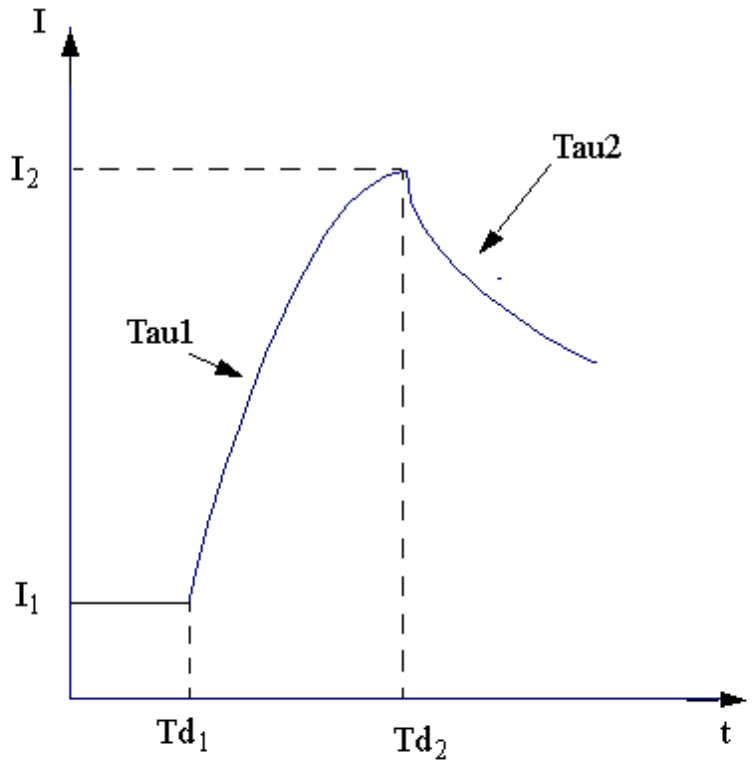
DC Current Source

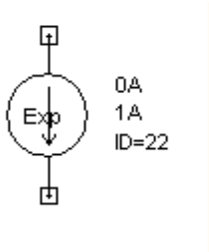
This type of source specifies the DC current flowing in a conductor. You can set either the total current or the current density flowing in the object.

- If total current is specified, the current density is assumed to be uniformly distributed throughout the object.
- If current density is specified, you may define a uniform current density or one that varies as a function of position.

Exponential Current Source

This is an independent current source with an exponential waveform of the current as function of time as shown in the following figure. The arrow symbol shows the direction of the positive current flow through the current source.



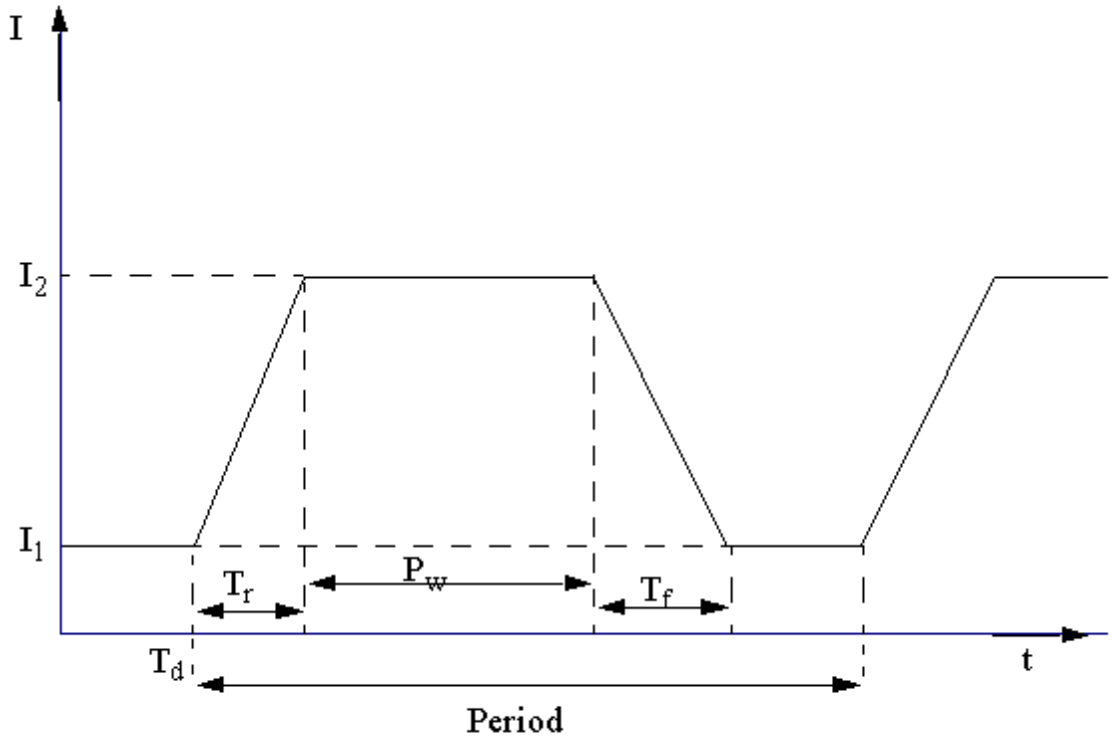


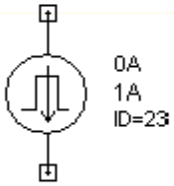
Thus, the parameters of this exponential source are:

- Initial current in Amps, I1;
- Peak current in Amps, I2;
- Rise time delay, Td1:
  - In seconds if type is TIME
  - In degrees if type is POSITION and type of motion is rotational;
  - In geometry units if type is POSITION and type of motion is translational;
  - In rpm if type is SPEED and type of motion is rotational;
  - In geometry units per second if type is SPEED and type of motion is translational;
- Rise time constant, Tau1:
  - In seconds if type is TIME
  - In degrees if type is POSITION and type of motion is rotational;
  - In geometry units if type is POSITION and type of motion is translational;
  - In rpm if type is SPEED and type of motion is rotational;
  - In geometry units per second if type is SPEED and type of motion is translational;
- Fall time delay, Td2:
  - In seconds if type is TIME
  - In degrees if type is POSITION and type of motion is rotational;
  - In geometry units if type is POSITION and type of motion is translational;
  - In rpm if type is SPEED and type of motion is rotational;
  - In geometry units per second if type is SPEED and type of motion is translational;
- Fall time constant, Tau2:
  - In seconds if type is TIME
  - In degrees if type is POSITION and type of motion is rotational;
  - In geometry units if type is POSITION and type of motion is translational;
  - In rpm if type is SPEED and type of motion is rotational;
  - In geometry units per second if type is SPEED and type of motion is translational;

Pulse Current Source

This is an independent current source with a trapezoidal waveform of the current as a function of time. The arrow symbol shows the direction of the positive current flow through the current source.





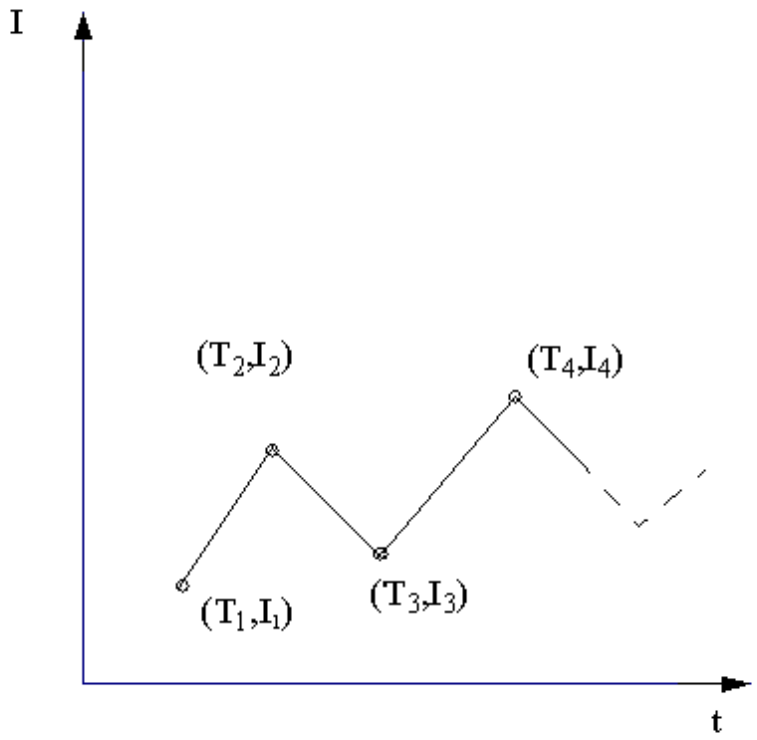
The parameters of a pulse current source are the following:

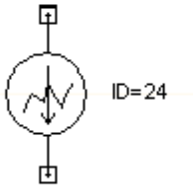
- Initial current in Amps, I1;
- Peak current in Amps, I2;
- Initial delay time, Td:
  - In seconds if type is TIME
  - In degrees if type is POSITION and type of motion is rotational;
  - In geometry units if type is POSITION and type of motion is translational;
  - In rpm if type is SPEED and type of motion is rotational;
  - In geometry units per second if type is SPEED and type of motion is translational;
- Rise time, Tr:
  - In seconds if type is TIME

    - In degrees if type is POSITION and type of motion is rotational;
    - In geometry units if type is POSITION and type of motion is translational;
    - In rpm if type is SPEED and type of motion is rotational;
    - In geometry units per second if type is SPEED and type of motion is translational;
- Fall time Tf:
    - In seconds if type is TIME
    - In degrees if type is POSITION and type of motion is rotational;
    - In geometry units if type is POSITION and type of motion is translational;
    - In rpm if type is SPEED and type of motion is rotational;
    - In geometry units per second if type is SPEED and type of motion is translational;
- Pulse width, Pw:
    - In seconds if type is TIME
    - In degrees if type is POSITION and type of motion is rotational;
    - In geometry units if type is POSITION and type of motion is translational;
    - In rpm if type is SPEED and type of motion is rotational;
    - In geometry units per second if type is SPEED and type of motion is translational;
- Pulse period, Period:
    - In seconds if type is TIME
    - In degrees if type is POSITION and type of motion is rotational;
    - In geometry units if type is POSITION and type of motion is translational;
    - In rpm if type is SPEED and type of motion is rotational;
    - In geometry units per second if type is SPEED and type of motion is translational;

Piecewise Linear Current Source

This is an independent current source with a piecewise linear waveform of the current as a function of time. The arrow symbol shows the direction of the positive current flow through the current source.





A piecewise linear current source is described by up to 20 pairs (Ti, Ii), where every pair of values specifies the value Ii in Amps of the current at time Ti in the following units:

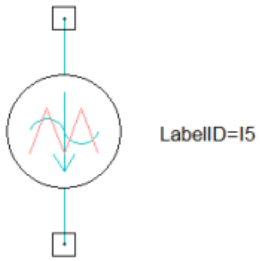
- In seconds if type is TIME.
- In degrees if type is POSITION and type of motion is rotational.
- In geometry units if type is POSITION and type of motion is translational.
- In rpm if type is SPEED and type of motion is rotational.
- In geometry units per second if type is SPEED and type of motion is translational.

PWM Current Source

The PWM current source component shown below provides PWM control signals used to support TDM parallel computation. (Refer to *Time Decomposition Method for Maxwell Transient Designs* in the Maxwell help.) The arrow symbol shows the direction of positive current flow through the current source.

Since all time steps in one subdivision along the time axis are solved together when TDM is used, all controlling signals in a control circuit cannot depend on solutions such as currents/voltages of the previous time step. Therefore, all diodes and switches which are controlled by independent voltages and currents are not supported.

In PWM control, the controlling signals are purely time and/or position dependent, therefore, PWM control is supported in TDM. In a PWM control circuit, since the upper and lower switches belong to the same phase are turned on and off at exactly same time, the free-wheeling diodes are not necessary. In order for users to directly use existing PWM control circuit, PWM current and voltage source components have been introduced to replace original controlling sub-circuit which includes a triangle current/voltage source connected in series with a sinusoidal current/voltage source. The circuit solver will detect all switches which are controlled by PWM current/voltage sources and automatically filter all related diodes, which means during the simulation, diodes are automatically excluded in PWM control circuit.

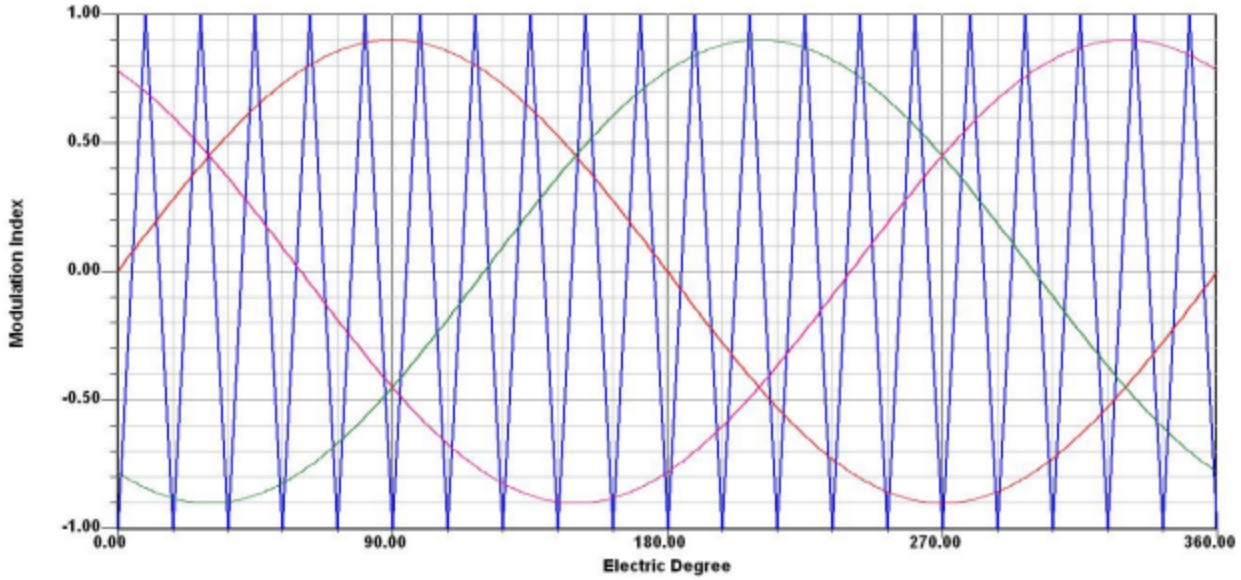


Note	Prior to the introduction of the PWM current source (IPWM) component, PWM control signals used to support TDM parallel computation were obtained from the difference of two current sources: a carrier signal with triangular waveform (IPulse), and a control signal with sinusoidal waveform (ISin)

An **IPWM** current source is defined by the following parameters:

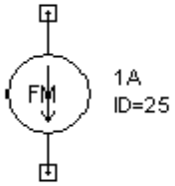
- **ModuIndex** is the modulation index, the ratio of control-signal amplitude to carrier-signal amplitude.
- **FreqTime** is the ratio of carrier-signal frequency to control-signal frequency.
- **IFreq** is the frequency of the control signal, if **Type** is **TIME**.
- **Phase** is the phase delay of the control signal, in electrical degrees.
- **Type** can be chosen as either **TIME** or **POS**, defining the output signal as a function of time or position.

The following figure shows typical three-phase PWM control signals.



Frequency-Modulated Sinusoidal Current Source

This is an independent current source with a single frequency modulated sinusoidal waveform of the current as a function of time. The arrow symbol shows the direction of the positive current flow through the current source.



The equation describing the waveform is:

$$I(t) = I_0 + I_a \cdot \sin[(2\pi \cdot F_C \cdot t) + M_{di} \cdot \sin(2\pi \cdot F_S \cdot t)]$$

where:

- Io is Offset current in Amps.
- Ia is the peak amplitude in Amps.
- FC is the carrier frequency if type is TIME.

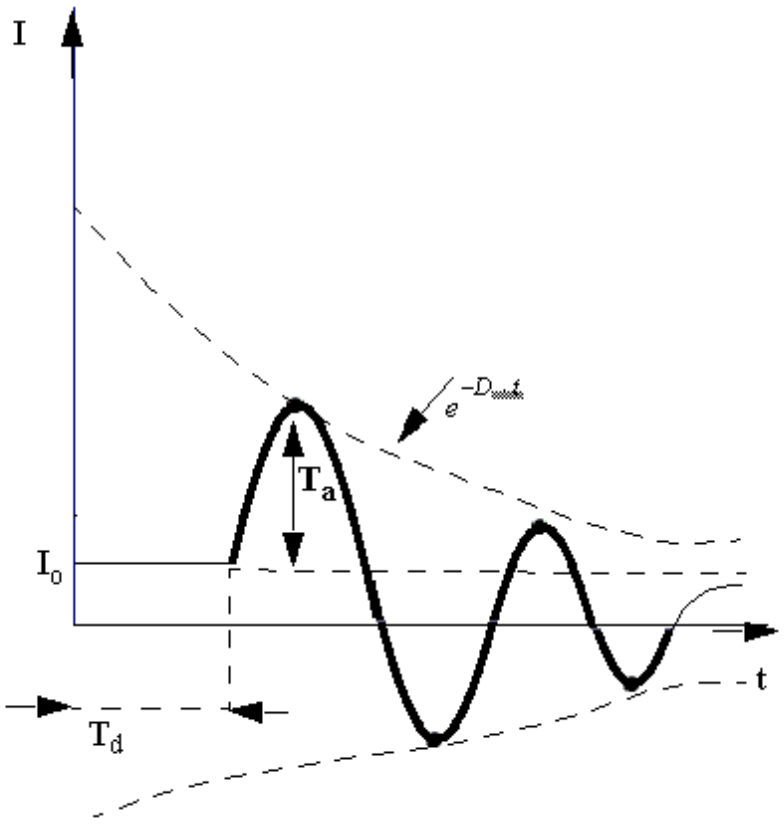
- Mdi is the modulation index.
- FS is the signal frequency if type is TIME.

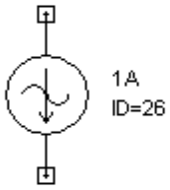
If the type is POSITION, the frequency should be calculated based on the respective spatial periodicity, taking into account the fact that "t" in the above equation is measured in degrees for rotational type of motion and in the user-defined geometry units for translational type of motion.

If the type is SPEED, the frequency should be calculated based on the respective speed periodicity, taking into account the fact that "t" in the above equation is measured in rpm for rotational type of motion and in the user-defined geometry units per second for translational type of motion.

Sinusoidal Current Source

This is an independent current source with an exponentially damped sinusoidal waveform of the current as a function of time. The arrow symbol shows the direction of the positive current flow through the current source.





The equation describing the waveform is:

$$I(t) = I_0 + I_a \cdot e^{-Df(t-T_d)} \cdot \sin[2 \cdot \pi \cdot IFreq \cdot (t-T_d) - Phase]$$

where:

- Io is Offset current in Amps.
- Ia is the peak amplitude in Amps.
- IFreq is the signal frequency if type is TIME.
- Td is the delay time in seconds if type is TIME.
- Phase is the signal phase delay if type is TIME.
- Df is the damping factor in 1/seconds if type is TIME.

If the type is POSITION, the frequency should be calculated based on the respective spatial periodicity, taking into account the fact that "t" in the above equation is measured in degrees for rotational type of motion and in the user-defined geometry units for translational type of motion. The delay and damping factor should also be interpreted accordingly.

If the type is SPEED, the frequency should be calculated based on the respective speed periodicity, taking into account the fact that "t" in the above equation is measured in rpm for rotational type of motion and in the user-defined geometry units per second for translational type of motion. The delay and damping factor should also be interpreted accordingly.

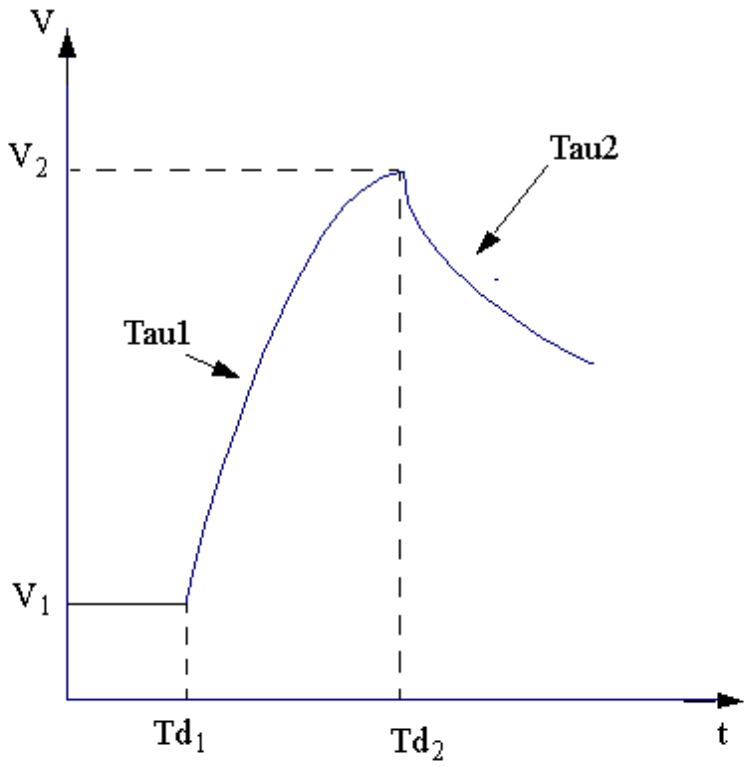
DC Voltage Source

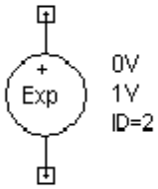
This type of source specifies the total DC voltage (electric potential) on a conductor. Voltages can be defined as constants or as math functions; however, the potential on a conductor is constant over the entire conductor. Note that conductors that touch should be set to the same voltage or defined as a single voltage source, since their potentials are identical.

DC voltage may also be specified on an edge or edges. Voltages can be defined as constant or as functions of position (for instance, to model a specific distribution of potential on the surface of a dielectric).

Exponential Voltage Source

This is an independent voltage source with an exponential waveform of the voltage as a function of time. The "+" and "-" symbols are used to mark the polarity of the source.





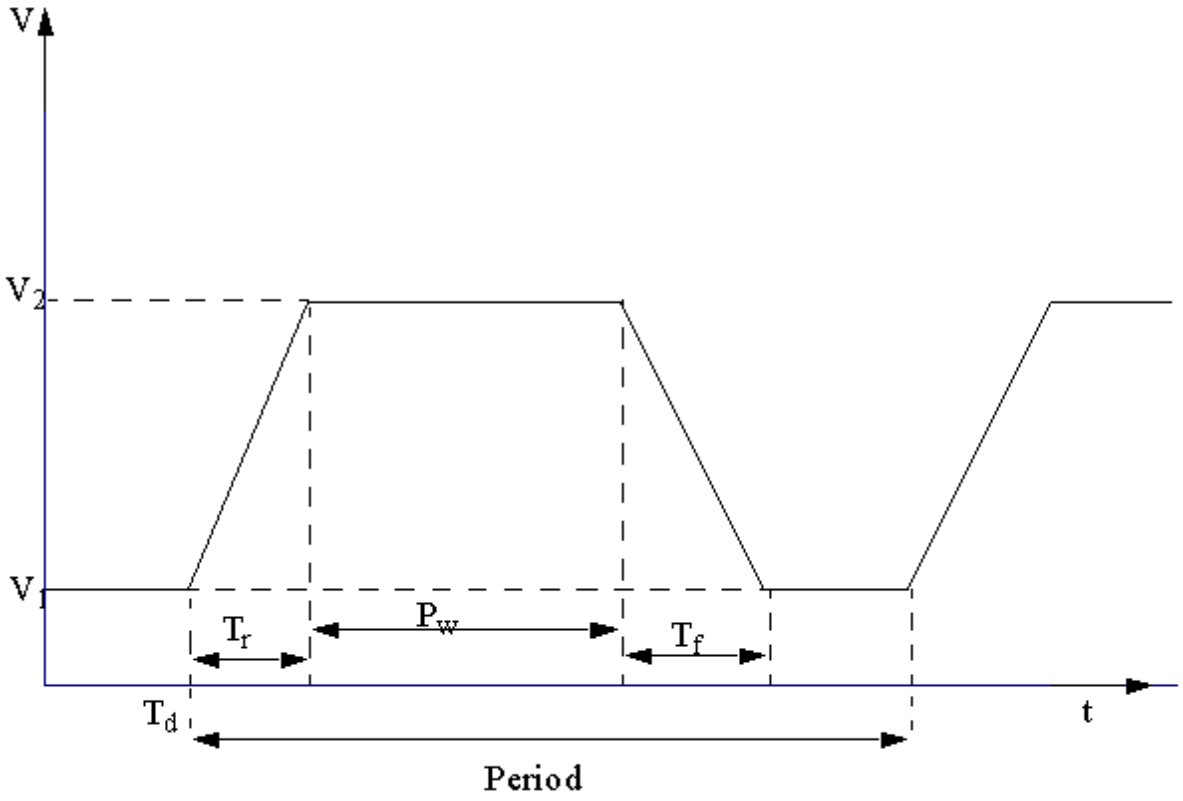
The parameters of an exponential voltage source are the following:

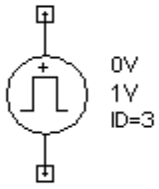
- Initial voltage in Volts, V1.
- Peak voltage in Volts, V2.
- Rise time delay, Td1:
  - In seconds if type is TIME
  - In degrees if type is POSITION and type of motion is rotational;

  - In geometry units if type is POSITION and type of motion is translational;
  - In rpm if type is SPEED and type of motion is rotational;
  - In geometry units per second if type is SPEED and type of motion is translational;
- Rise time constant, Tau1:
  - In seconds if type is TIME
  - In degrees if type is POSITION and type of motion is rotational;
  - In geometry units if type is POSITION and type of motion is translational;
  - In rpm if type is SPEED and type of motion is rotational;
  - In geometry units per second if type is SPEED and type of motion is translational;
- Fall time delay, Td2:
  - In seconds if type is TIME
  - In degrees if type is POSITION and type of motion is rotational;
  - In geometry units if type is POSITION and type of motion is translational;
  - In rpm if type is SPEED and type of motion is rotational;
  - In geometry units per second if type is SPEED and type of motion is translational;
- Fall time constant, Tau2:
  - In seconds if type is TIME
  - In degrees if type is POSITION and type of motion is rotational;
  - In geometry units if type is POSITION and type of motion is translational;
  - In rpm if type is SPEED and type of motion is rotational;
  - In geometry units per second if type is SPEED and type of motion is translational;

Pulse Voltage Source

This is an independent voltage source with a trapezoidal waveform of the voltage as a function of time. The "+" and "-" symbols are used to mark the polarity of the source.





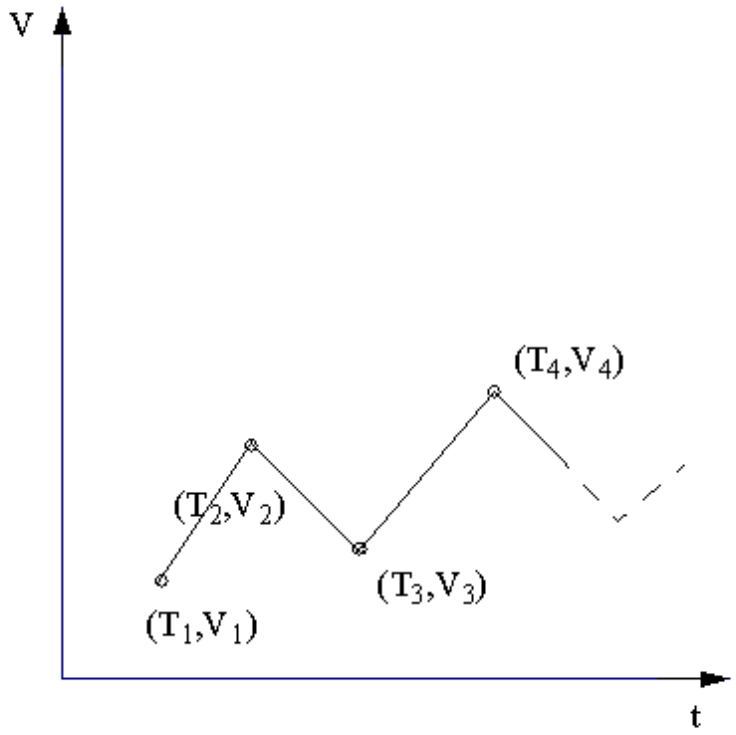
The parameters of a voltage pulse source are the following:

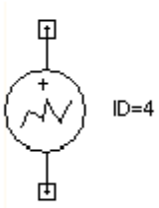
- Initial voltage in Volts, V1.
- Peak voltage in Volts, V2.
- Initial delay time, Td:
  - In seconds if type is TIME
  - In degrees if type is POSITION and type of motion is rotational;
  - In geometry units if type is POSITION and type of motion is translational;
  - In rpm if type is SPEED and type of motion is rotational;
  - In geometry units per second if type is SPEED and type of motion is translational;

- Rise time, Tr:
  - In seconds if type is TIME
  - In degrees if type is POSITION and type of motion is rotational;
  - In geometry units if type is POSITION and type of motion is translational;
  - In rpm if type is SPEED and type of motion is rotational;
  - In geometry units per second if type is SPEED and type of motion is translational;
- Fall time Tf:
  - In seconds if type is TIME
  - In degrees if type is POSITION and type of motion is rotational;
  - In geometry units if type is POSITION and type of motion is translational;
  - In rpm if type is SPEED and type of motion is rotational;
  - In geometry units per second if type is SPEED and type of motion is translational;
- Pulse width, Pw:
  - In seconds if type is TIME
  - In degrees if type is POSITION and type of motion is rotational;
  - In geometry units if type is POSITION and type of motion is translational;
  - In rpm if type is SPEED and type of motion is rotational;
  - In geometry units per second if type is SPEED and type of motion is translational;
- Pulse period, Period:
  - In seconds if type is TIME
  - In degrees if type is POSITION and type of motion is rotational;
  - In geometry units if type is POSITION and type of motion is translational;
  - In rpm if type is SPEED and type of motion is rotational;
  - In geometry units per second if type is SPEED and type of motion is translational;

Piecewise Linear Voltage Source

This is an independent voltage source with a piecewise linear waveform of the voltage as a function of time. The "+" and "-" symbols are used to mark the polarity of the source.





A piecewise linear voltage source is described by up to 20 pairs (Ti, Vi), where every pair of values specifies the value Vi in Volts of the voltage at time Ti in the following units:

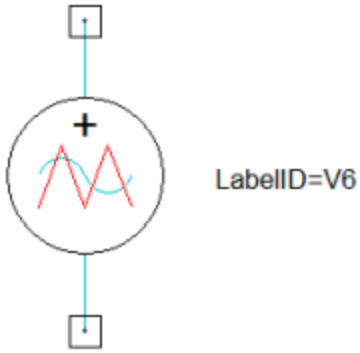
- In seconds if type is TIME.
- In degrees if type is POSITION and type of motion is rotational.
- In geometry units if type is POSITION and type of motion is translational.
- In rpm if type is SPEED and type of motion is rotational.
- In geometry units per second if type is SPEED and type of motion is translational.

PWM Voltage Source

The PWM voltage source component shown below provides PWM control signals used to support TDM parallel computation. (Refer to *Time Decomposition Method for Maxwell Transient Designs* in the Maxwell help.) The "+" symbol is used to mark the polarity of the source.

Since all time steps in one subdivision along the time axis are solved together when TDM is used, all controlling signals in a control circuit cannot depend on solutions such as currents/voltages of the previous time step. Therefore, all diodes and switches which are controlled by independent voltages and currents are not supported.

In PWM control, the controlling signals are purely time and/or position dependent, therefore, PWM control is supported in TDM. In a PWM control circuit, since the upper and lower switches belong to the same phase are turned on and off at exactly same time, the free-wheeling diodes are not necessary. In order for users to directly use existing PWM control circuit, PWM current and voltage source components have been introduced to replace original controlling sub-circuit which includes a triangle current/voltage source connected in series with a sinusoidal current/voltage source. The circuit solver will detect all switches which are controlled by PWM current/voltage sources and automatically filter all related diodes, which means during the simulation, diodes are automatically excluded in PWM control circuit.

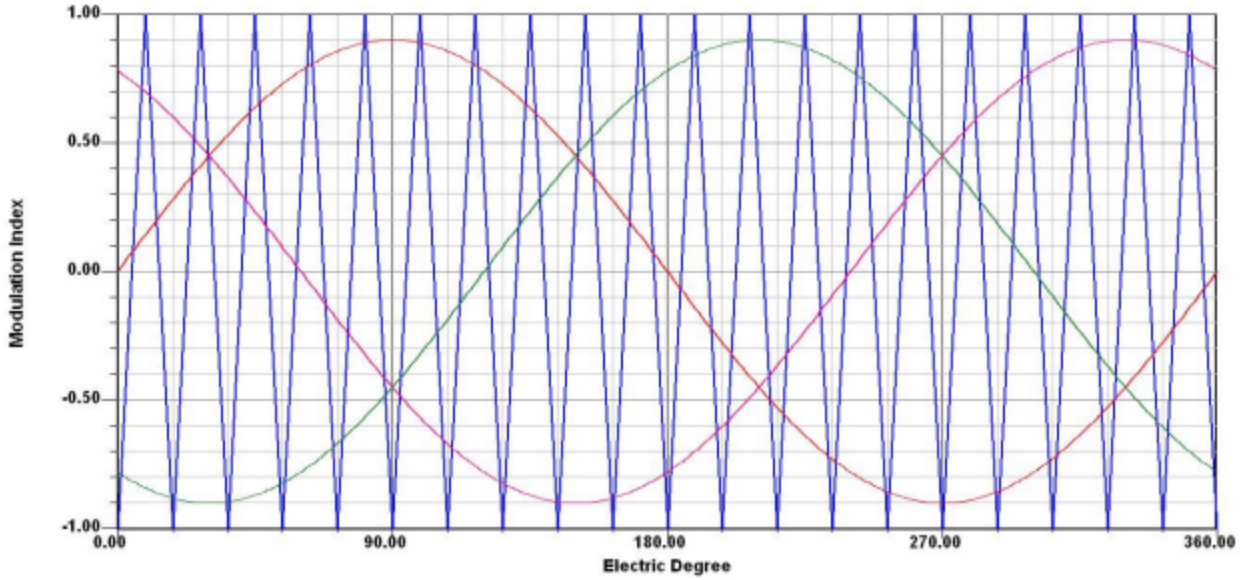


Note	Prior to the introduction of the PWM voltage source (VPWM) component, PWM control signals used to support TDM parallel computation were obtained from the difference of two voltage sources: a carrier signal with triangular waveform (VPulse), and a control signal with sinusoidal waveform (VSin)

A **VPWM** source is defined by the following parameters:

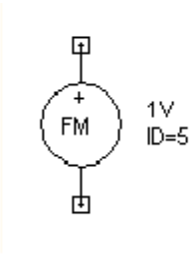
- **ModuIndex** is the modulation index, the ratio of control-signal amplitude to carrier-signal amplitude.
- **FreqTime** is the ratio of carrier-signal frequency to control-signal frequency.
- **VFreq** is the frequency of the control signal, if **Type** is **TIME**.
- **Phase** is the phase delay of the control signal, in electrical degrees.
- **Type** can be chosen as either **TIME** or **POS**, defining the output signal as a function of time or position.

The following figure shows typical three-phase PWM control signals.



Frequency-Modulated Sinusoidal Voltage Source

This is an independent voltage source with a single frequency modulated sinusoidal waveform of the voltage as a function of time. The "+" and "-" symbols are used to mark the polarity of the source.



The equation describing the waveform is:

$$V(t) = V_0 + V_a \cdot \sin[(2\pi \cdot F_C \cdot t) + M_{di} \cdot \sin(2\pi \cdot F_S \cdot t)]$$

where:

- Vo is Offset voltage in Volts.
- Va is the peak amplitude in Volts.
- FC is the carrier frequency if type is TIME.

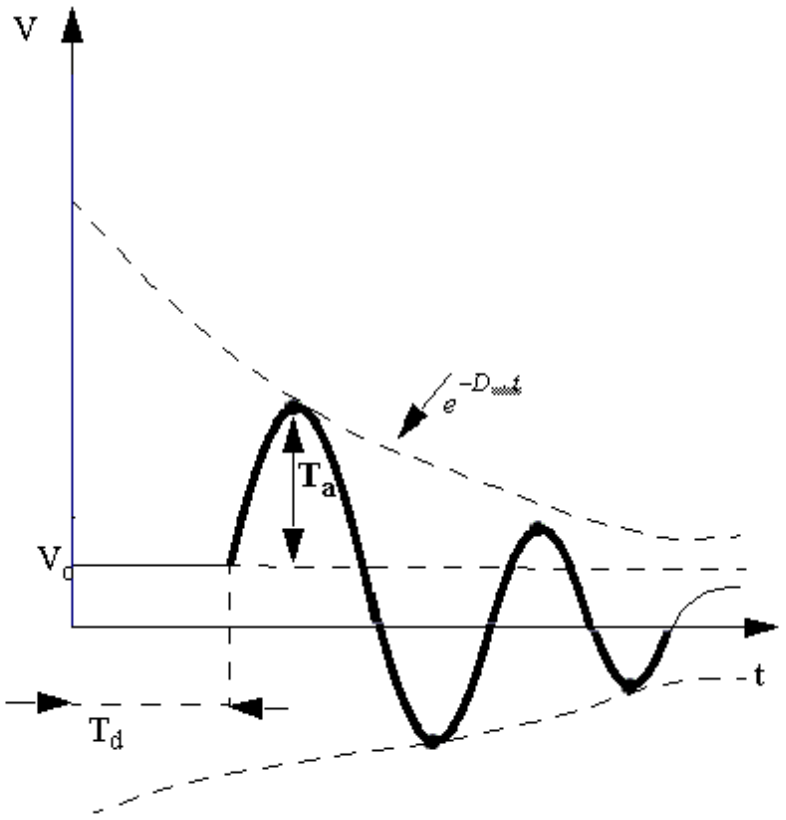
- Mdi is the modulation index.
- FS is the signal frequency if type is TIME.

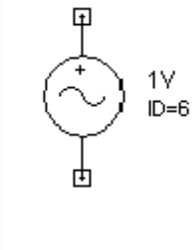
If the type is POSITION, the frequency should be calculated based on the respective spatial periodicity. taking into account the fact that "t" in the above equation is measured in degrees for rotational type of motion and in the user-defined geometry units for translational type of motion.

If the type is SPEED, the frequency should be calculated based on the respective speed periodicity, taking into account the fact that "t" in the above equation is measured in rpm for rotational type of motion and in the user-defined geometry units per second for translational type of motion.

Sinusoidal Voltage Source

This is an independent voltage source with an exponentially damped sinusoidal waveform of the voltage as a function of time. The "+" and "-" symbols are used to mark the polarity of the source.





The equation describing the waveform is:

$$V(t) = V_0 + V_a \cdot e^{-Df(t-T_d)} \cdot \sin[2 \cdot \pi \cdot VFreq \cdot (t-T_d) - Phase]$$

where:

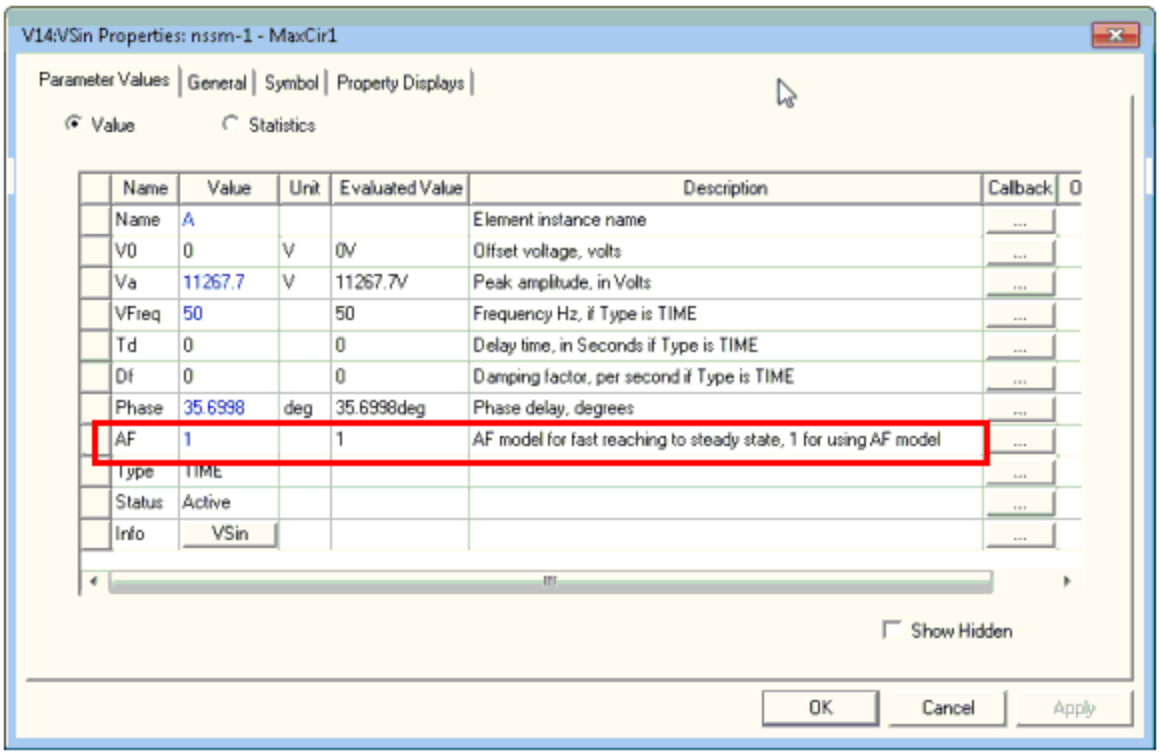
- Vo is Offset voltage in Volts.
- Va is the peak amplitude in Volts.
- VFreq is the signal frequency if type is TIME.
- Td is the delay time in seconds if type is TIME.
- Phase is the signal phase delay if type is TIME.
- Df is the damping factor in 1/seconds if type is TIME.

If the type is POSITION, the frequency should be calculated based on the respective spatial periodicity, taking into account the fact that "t" in the above equation is measured in degrees for rotational type of motion and in the user-defined geometry units for translational type of motion. The delay and damping factor should also be interpreted accordingly.

If the type is SPEED, the frequency should be calculated based on the respective speed periodicity, taking into account the fact that "t" in the above equation is measured in rpm for rotational type of motion and in the user-defined geometry units per second for translational type of motion. The delay and damping factor should also be interpreted accordingly.

Using the AF (alternating flux) model

For transient simulations whose excitation is the sinusoidal voltage source, the DC flux linkage component may take a very long time to decay, especially for a device with a large time constant. Setting AF=1 enables the AF model to be used by the Maxwell transient solver.



The AF model algorithm adds an additional voltage component to the original voltage definition during the first half-cycle to quickly eliminate the DC flux linkage, thus speeding up simulation solutions.

Note	• RMxprt automatically enables the AF model in the voltage definition for each winding when the Maxwell 2D or 3D model and associated Maxwell circuit are created. • This speedup capability is only applicable to voltage sources whose integration of voltage quantity over one period is zero. This means the DC component of all voltage sources must also be zero.

Placing Components in the Maxwell Circuit Editor Schematic

To place a circuit component:

1. Click the **Components** tab in the **Component Libraries Window**.
2. In the tree, expand the **Maxwell Circuit Elements** branch.
3. Expand the branch containing the component you want to place. The choices are **Dedicated Elements**, **Passive Elements**, **Probes**, and **Sources**.
4. Click to select the name of the component you want to place.
5. Drag the component to the Schematic window.
   A diagram of the component appears connected to the mouse pointer.

6. Release the mouse button to place the component in the location you prefer.
7. To place a second component of the same type, move the mouse pointer to another location, and release it again.
8. To exit from placement mode, do one of the following
   - To place the component a final time before exiting, press **ENTER**.
   - To exit without placing the component again, press **SPACE**.

Hint	To access these commands, you can also right-click and select one of the following from the shortcut menu: • Place and Finish • Finish

Related Topics

Assigning Component Properties in Maxwell Circuit Editor

Assigning Component Properties in Maxwell Circuit Editor

Once a component has been placed in the Schematic window, do the following to edit its properties:

1. Double-click the component.
   The **Properties** window opens.
2. Make the desired edits on the following four tabs:
   - Parameter Values
   - General
   - Symbol
   - Property Displays

Note	Variables may be assigned to parameter values in the circuit by entering a variable name in the parameter value field and assigning a value in the **Add Variable** dialog box. Variables assigned in **Maxwell Circuit Editor** will be exported with the circuit and are available for modifying when the circuit is imported into another Ansys application.

3. Optionally, you can select or clear the **Show Hidden** check box on any of the **Properties** window tabs.
4. Click **OK**.

The specific properties differ per component, as seen in the help topics describing each circuit component.

Each tab of the **Properties** window contains a list of parameters (each individual row) and properties you need to set for each parameter (each column is a property).

Related Topics

Placing Components in the Maxwell Circuit Editor Schematic
Callback Scripting Using PropHost Object

Callback Scripting Using PropHost Object

Callback scripts are scripts that can be set in the Property Dialog for individual properties by clicking the button in the Callback column and choosing a script that is saved with the project. Callback scripts can contain any legal script commands including general Ansys script function calls (e.g. GetApplication(), …). In addition, they can call functions on a special object named PropHost. The PropHost represents the PropServer (owner of properties) that contains the Property that is calling the Callback script. Therefore, the Callback script can use the PropHost's functions to query or set other properties in the same PropServer. Refer to the Property Level scripting commands for more information.

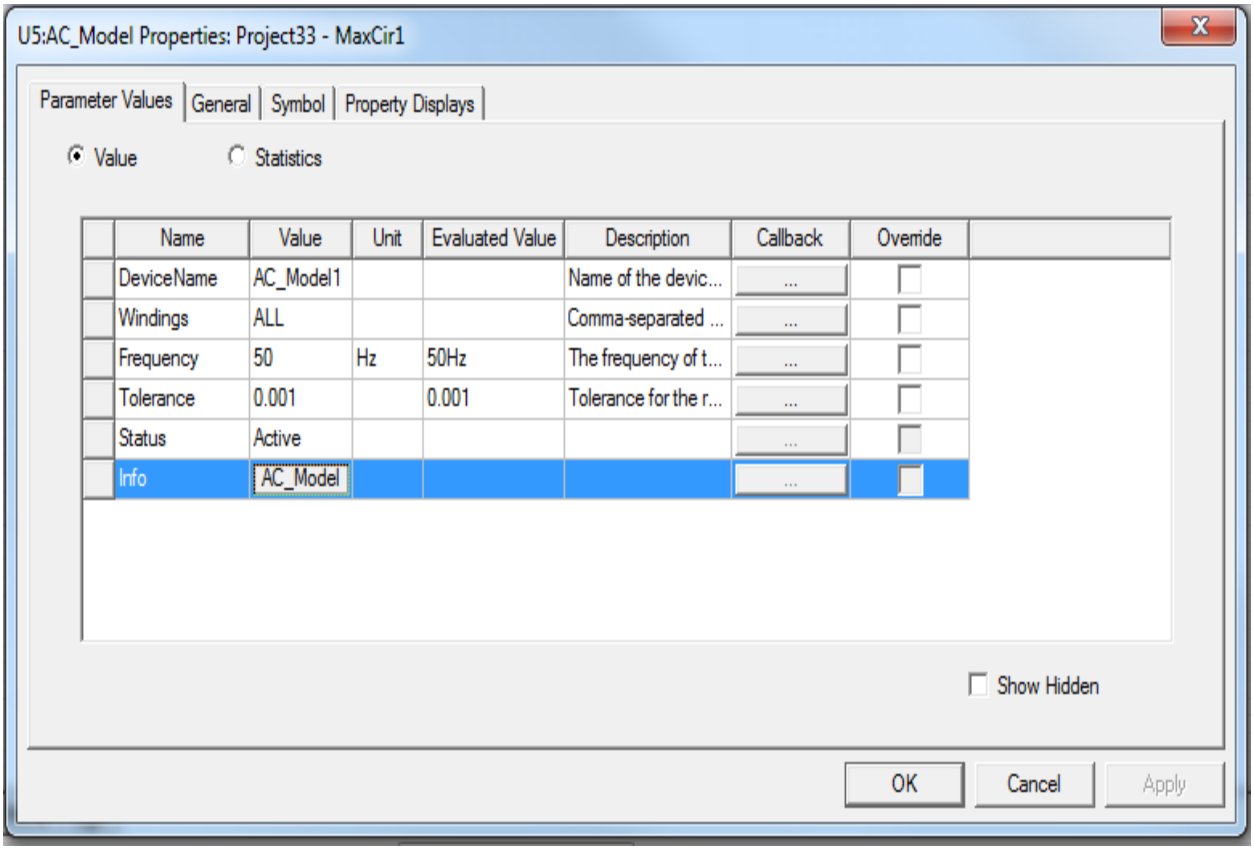
Related Topics

Scripting Guide: *Property Level Commands*

Opening the Help for Circuit Components

To start the help for any component from the schematic editor:

1. In the Schematic editor, double-click the component for which you want to view help.
   The **Properties** dialog box opens.
2. Click the **Parameter Values** tab.
3. Select the **Value** radio button.
4. In the **Info** row, click the button in the **Value** column, as shown below for **AC_Model**:



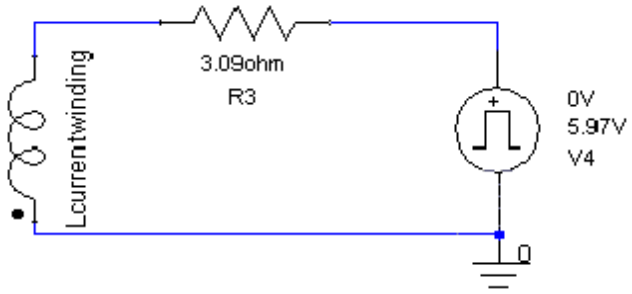
The help viewer opens to display the component's specifications.

Setting Up an External Circuit Using the Maxwell Circuit Editor

Note	One use for an external circuit can be to supply an excitation to a coil terminal, rather than using a voltage type of excitation.

Note	External circuits intended for use with the Time Decomposition Method (TDM) can only support basic circuit elements that are solution-independent. This is because, for TDM, all time steps (or all time steps in a subdivision) are solved simultaneously - and thus the circuit topology and its parameters must be known before solving. Therefore some circuit elements, such as solution-dependent switches, diodes, capacitors, and inductors which are not directly connected to windings in series, are not supported.

The driving circuit for the winding in this design consists of a voltage source in series with a resistor and with the winding. When complete, the circuit should look similar to the following figure.



To set up the external circuit, follow this general procedure:

1. Add the circuit elements.
2. Connect the circuit elements in series.
3. Export the netlist.
4. Save the Maxwell Circuit Editor project.
5. Assign the external circuit.

Add the Circuit Elements

To add the circuit elements in Maxwell Circuit Editor:

1. Click **Project>Insert Maxwell Circuit Design**.
   The circuit sheet appears.
2. Place the winding circuit element on the sheet:
   a. In the **Component Libraries**, under **Maxwell Circuit Elements>Dedicated Elements**, select the **Winding** element.
   b. Drag it onto the sheet.
   c. Right-click, and select **Finish** to place the component.
   d. To view the properties, double-click the component in the Schematic window.
   The **Properties** window appears.
   e. Change the **Name** to **currentwinding**, the same name you used when defining the winding in the Maxwell design.
   f. Click **OK**.
   g. Click **Draw>Rotate**, and position the winding vertically.
3. Place a resistor on the sheet:
   a. In the project tree, under **Passive Elements**, select **Resistor**.
   b. Drag the resistor onto the sheet.
   c. Right-click, and select **Finish** to place it where desired.
   d. Double-click the symbol of the resistor, change the value of the resistor, **R**, to **3.09**, keep the **Unit** value set to **ohm**, and click **OK**. The default is 100 Ohms.
4. Place a voltage pulse on the sheet:

a. In the project tree, under **Sources** select a **VPulse** element (Pulse Voltage Source).
b. Drag it to the sheet, and then right-click and select **Finish** to place it onto the sheet.
c. Double-click the source element symbol on the sheet, and then specify the following source characteristics:

Parameter	Value	Description
V1	0	Initial voltage
V2	5.97	Peak voltage
Tr	0.001	Rise time
Tf	0.001	Fall time
Pw	1	Pulse width
Period	2	

d. Leave the other fields set to the default values, and click **OK**.

Connect the Circuit Elements in Series

To connect the circuit elements in series:

1. From within the Maxwell Circuit Editor, click **Draw>Wire**.
2. Click each terminal.
3. When done, place the Ground symbol: Click **Draw>Ground**, place the **Ground** symbol on the sheet, right-click, and select **Finish** to place the symbol.
4. Connect the ground to the circuit: Click **Draw>Wire**, and draw the final wire.

Export the Netlist

To export the netlist:

1. From within the Maxwell Circuit Editor, click **Maxwell Circuit>Export Netlist**.
   The **Netlist Export** dialog box appears.

Note	To view the netlist before exporting it, click **Maxwell Circuit>Browse Netlist**.

2. Select the folder where you want to save the external circuit file.
3. Type a name for the circuit in the **File name** box.
4. Click **Save**.
   The **Netlist Export** dialog box closes and the Maxwell Circuit Editor reappears.

Save the Maxwell Circuit Editor Project

To save the project and exit Maxwell Circuit Editor:

1. Click **File>Save**, type a name for the project, and click **Save** to save the Maxwell Circuit Editor project.
2. Click **File>Exit** to close the Maxwell Circuit Editor program.

Assign the External Circuit

To assign the circuit in the Maxwell design:

1. Click **Maxwell2D** (or **3D**)**>Excitations>External Circuit>Edit External Circuit**.
   The **Edit External Circuit** dialog box appears.
2. Click **Import Circuit Netlist**.
   The **Select File** dialog box appears.
3. Select **Designer Net List Files (*.sph)** from the **Files of type** pull-down list.
4. Browse to the location where you saved the circuit, select it, and click **Open** to import it.
5. Click **OK** to close the **Edit External Circuit** dialog box.

Renaming a Source in Maxwell Circuit Editor

To rename a source that is drawn in the Schematic:

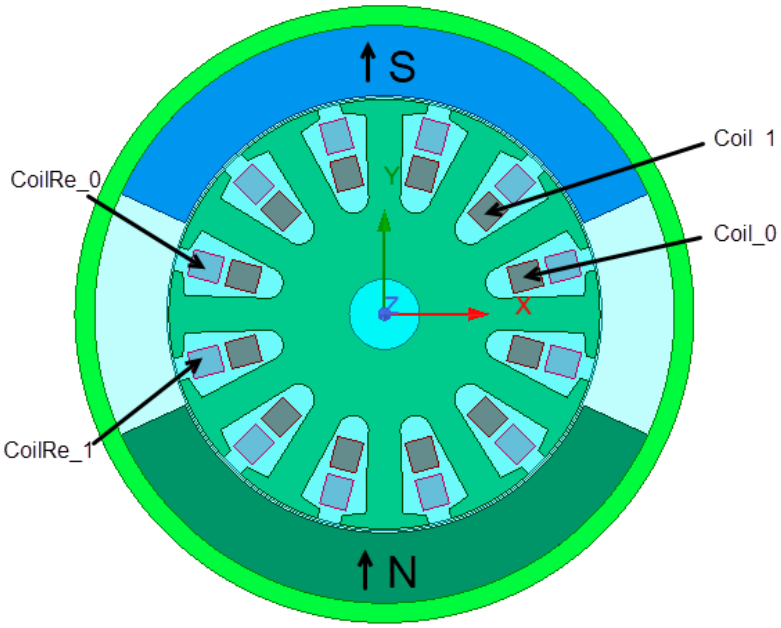
1. Double-click the source.
   The **Properties** dialog box appears.
2. Click the **Parameter Values** tab (which should be the default tab visible).
3. Change the text value in the **Name** row.
4. Click **OK**.
   The new name appears for the source label in the Schematic.

Applying the Commutating Bar Element

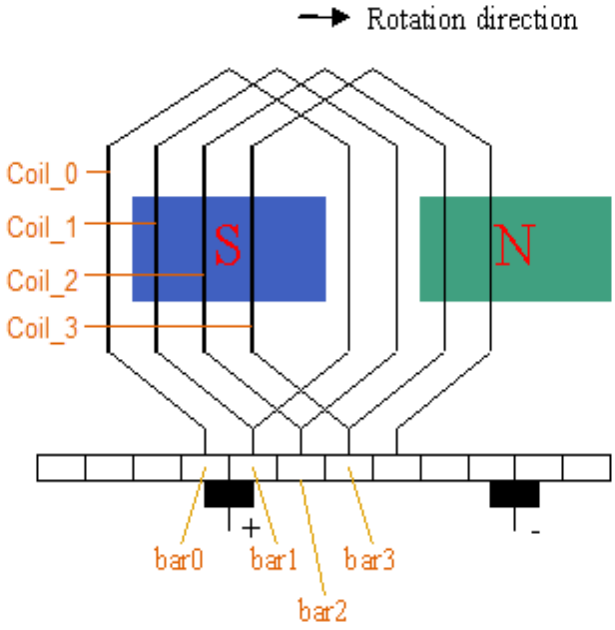
This paragraph describes how to apply the commutating bar element to simulate the commutating process of brush-type DC machines. A two-pole 12-slot PMDC motor (shown below) is used as an example.

Note	The example used here is based on the **stndj-1.aedt** project found in the *<install_dir>*/Examples/RMxprt/pmdc folder.

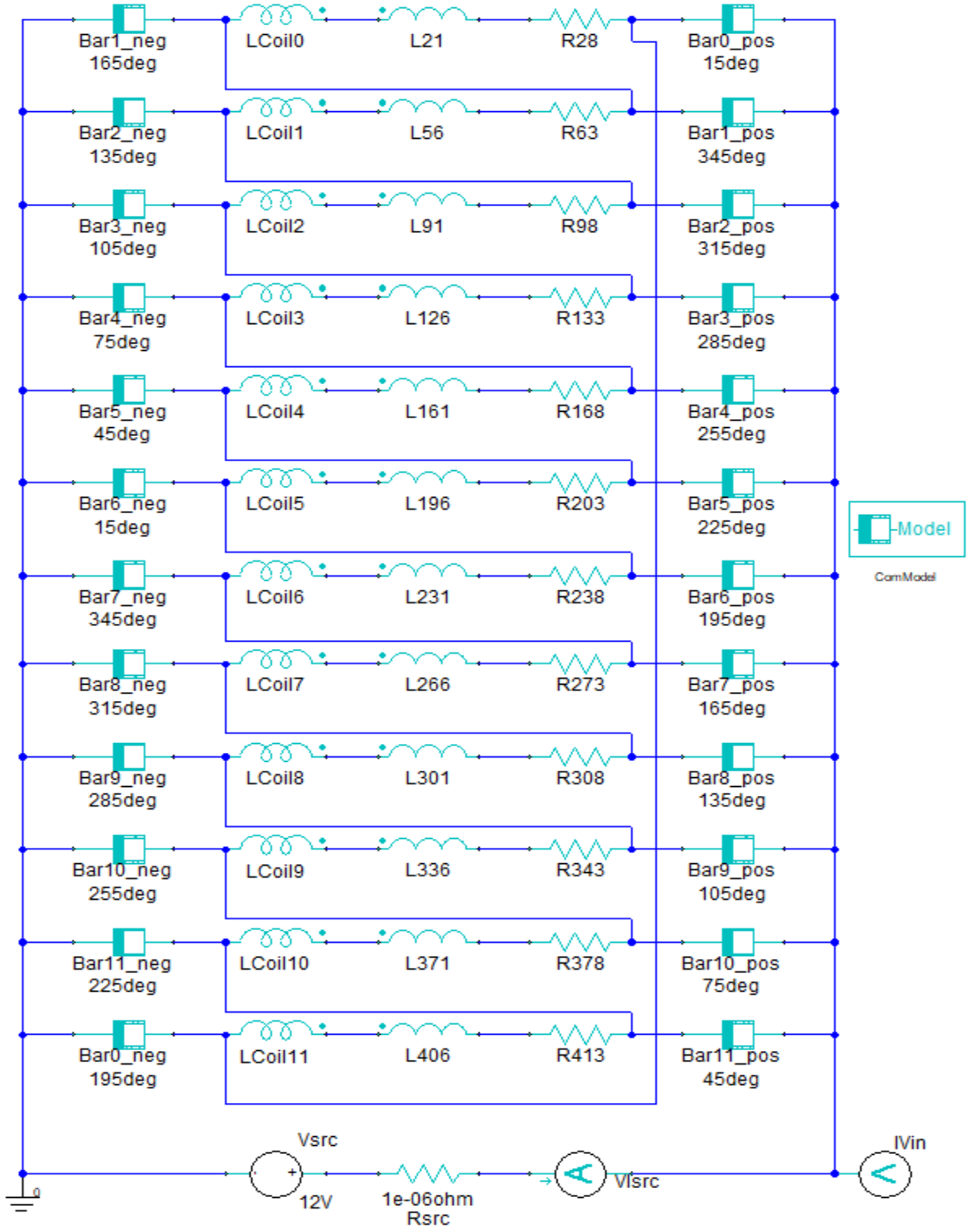
The flux direction inside the S pole permanent magnet is from the air gap to stator yoke, and that of the N pole is from the stator yoke to the air gap.



The DC winding is of lap type with coil pitch of 5 slots. The flat-out extensional drawing of the motor indicating the relationship of permanent magnets, coils, commutating bars, and brushes at the initial position is shown below. With the rotation direction shown, the brush aligned with the S pole is positive, and the brush aligned with the N pole is negative. The "go" terminal of Coil_0 connects to bar0, and its return terminal connects to bar1; the "go" terminal of Coil_1 connects to bar1, and its return terminal connects to bar2; and so on.

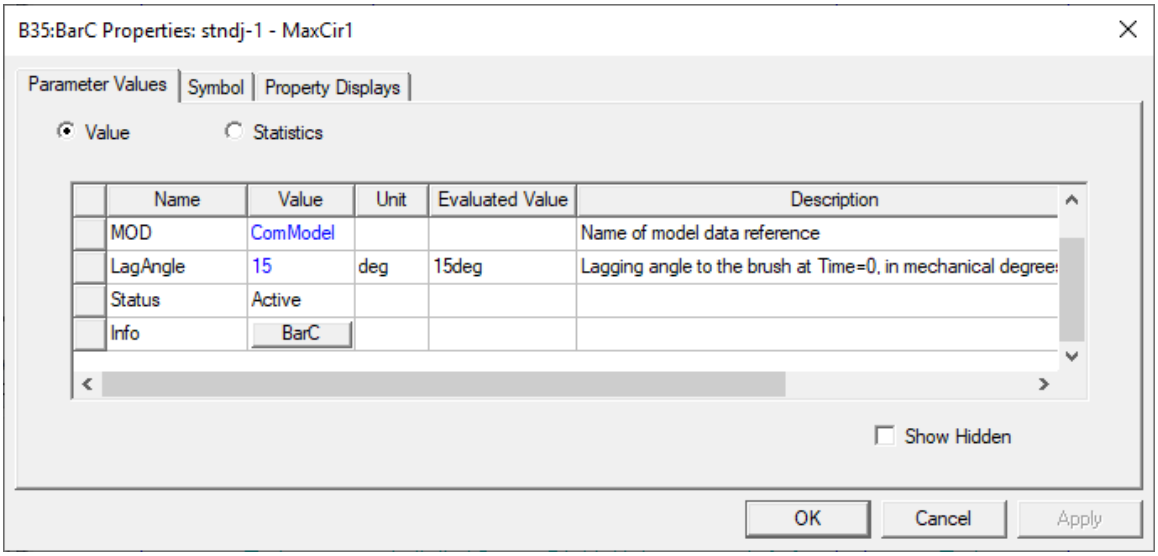


At the initial time, bar0 lags the positive brush by 15 mechanical degrees (a half commutating bar pitch), and it lags the negative brush by 195 mechanical degrees; bar1 lags the positive brush by -15 mechanical degrees, or 345 degrees, and it lags the negative brush by 165 mechanical degrees; and so on. The functional connection of each commutating bar with the positive brush or negative brush is modeled by BarC (commutating bar) elements, as shown below.

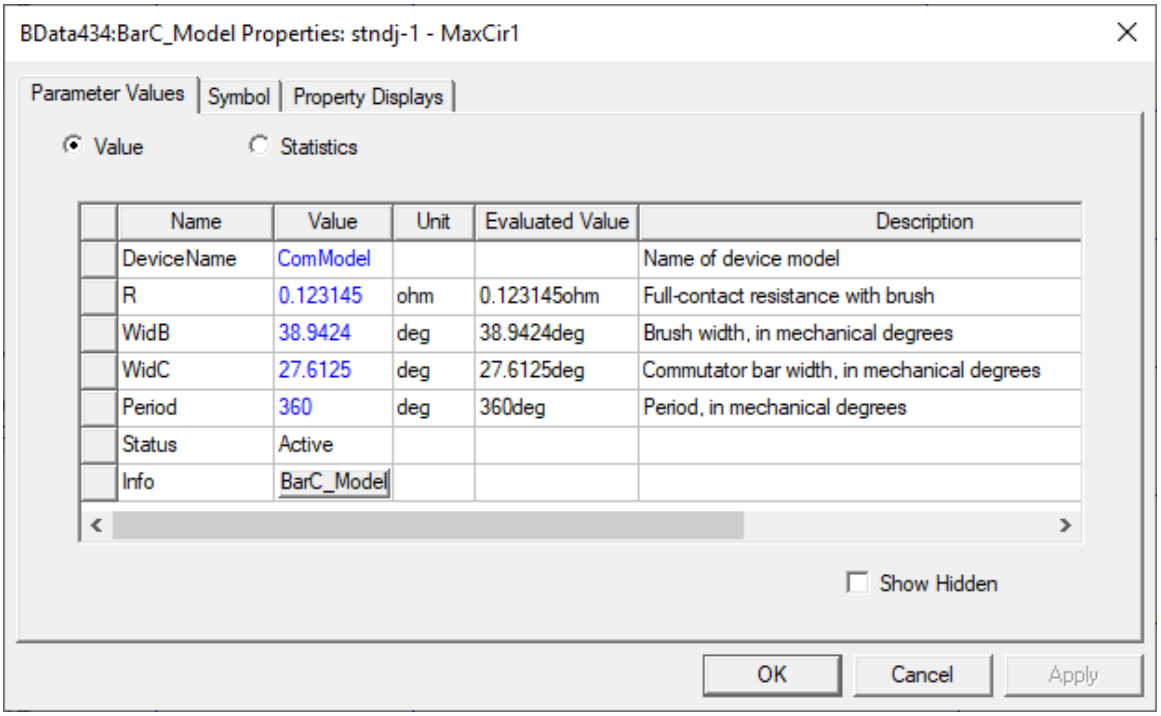


Elements Bar0_pos through Bar11_pos represent the functional connections of bar0 to bar11 with the positive brush, respectively. Elements Bar0_neg through Bar11_neg give the functional

connections of bar1 to bar11, and bar0 with the negative brush, respectively. The additional inductor in series with each coil inductance represents the end turn effect that is desirable to consider in a 2D model and possibly also in a 3D model that does not include the end turn geometry. The additional series resistor in series with each coil represents the global resistance of the coil and needs to be included in both 2D and 3D simulations. The values below the BarC element symbols are the respective lagging angles in mechanical degrees. The element parameters can be edited by double-clicking on the element, as shown below.



The input value "ComModel" for MOD is the model name that defines the parameters of commutating bars and brushes. The parameters of ComModel can be edited by double-clicking the BarC_Model element.



Note	As a rule, each BarC element references a unique BarC_Model element.

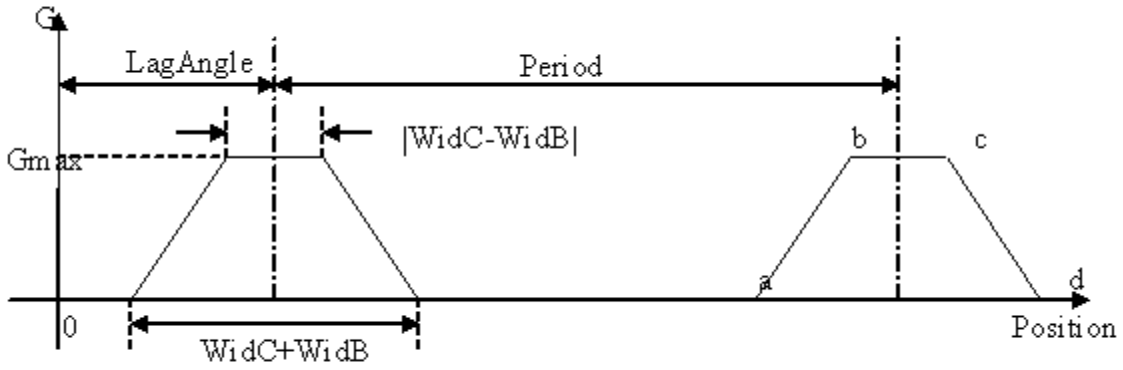
In this design, the commutator diameter is 24mm, and the brush width is 8mm. Therefore, the brush width in mechanical degrees is the following:

*WidB = 2 * arcsin(8/24) = 38.9 (deg)*

The number of commutator bars is 12 (the same as the number of slots), and the commutator insulation thickness is 0.5mm. Therefore, the commutator bar width in mechanical degrees is the following:

*WidC = 360/12 - 2 * arcsin(0.5/24) = 27.6 (deg)*

The electric conductance between a commutating bar and a brush varies with the rotor position and can be derived from the model parameters as shown below, where *Gmax = 1 / R*.



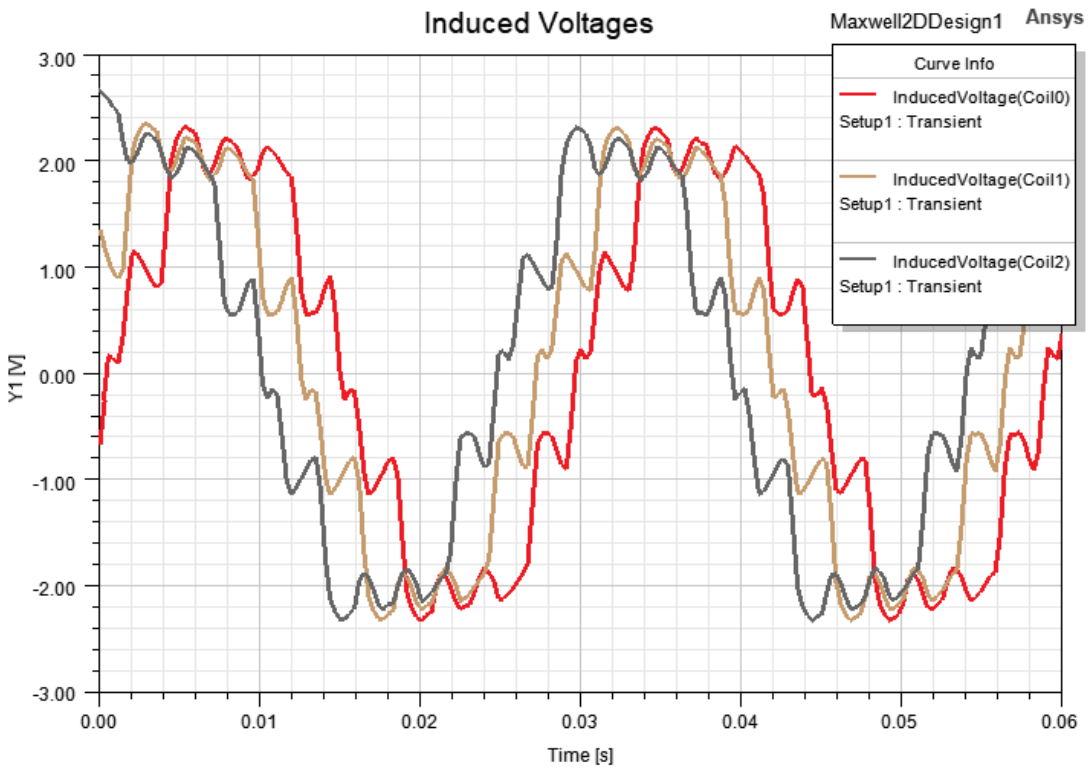
Electric conductance as a function of rotor position

LagAngle is treated as a BarC element parameter because it is different for different commutating bars. All other values (Gmax, WidB, WidC, and Period) can be obtained from BarC_Model parameters. Positions a, b, c, and d correspond to the positions when one side of a commutating bar (solid color) aligns to one side of a brush, as shown below.

Different conducting position when WidB > WidC

Different conducting position when WidB < WidC

The computed no-load induced voltages for three of the coils, taking into account the commutating process, are shown below.



28 - Maxwell 2D Technical Notes

Maxwell® 2D is an interactive software package that uses the finite element method (FEM) analysis to simulate and solve two-dimensional electromagnetic field problems. Maxwell integrates with other Ansys software to perform complex tasks while remaining easy to use.

Maxwell 2D is used for analyzing electric and magnetic fields in structures with uniform cross-sections or full rotational symmetry — where the field patterns in the entire device can be analyzed by modeling the field patterns in its cross-section.

In general, 2D modeling of electromagnetic fields solves for the fields in a 2D cross-section of a device which is considered to be placed far away from the extremities of the devices, thus ignoring end effects. While it can provide excellent accuracy for all applicable modeling tasks, 2D solution sequences cannot be always used to accurately model electromagnetic devices. Devices where end effects are significant or that may not exhibit complete rotational symmetry must be modeled in 3D for greatest accuracy.

Technical Notes

These technical notes contain background information on the theory behind the Maxwell, including:

- A list of all Maxwell software modules.
- Discussion of: background theory; capacitance, inductance, impedance, conductance, and admittance matrices; virtual force and torque; flux linkage; and current flow for the following field solvers:
  - Electrostatic
  - Magnetostatic
  - Eddy current
  - DC conduction
  - AC conduction
  - Transient
- The differences between field solutions in cartesian (XY) and axisymmetric (RZ) models.
- An overview of the phasor notation used in this guide to represent complex field quantities.

Related Topics

Optimization Overview

Solvers

Maxwell 2D consists of the following solver modules:

Electrostatic Field Solver	Computes the static electric field that exists in a structure given a distribution of DC voltages and static charges. A capacitance matrix, force, torque, and flux linkage may also be computed from the electric field.

Magnetostatic Field Solver	Computes the static magnetic field that exists in a structure given a distribution of DC currents and permanent magnets. The magnetic field may be computed in structures with both nonlinear and linear materials. An inductance matrix, force, torque, and flux linkage may also be computed from the energy stored in the magnetic field.
AC Conduction Field Solver	Computes the AC voltages and current density distribution in a material having both conductive and dielectric properties given a distribution of AC voltages. An admittance matrix and current flow may also be computed from the calculated fields.
DC Conduction Field Solver	Computes the DC currents that flow in a lossy dielectric given a distribution of DC voltages. A conductance matrix and current flow may also be computed from the computed electric field solution.
Eddy Current Field Solver	Computes the oscillating magnetic field that exists in a structure given a distribution of AC currents. Also computes current densities, taking into account all eddy current effects (including skin effects). An impedance matrix, force, torque, core loss, and current flow may also be computed from the computed field solution.
Transient Magnetic Solver	Computes transient (Time Domain) magnetic fields caused by permanent magnets, conductors, and windings supplied by voltage and/or current sources with arbitrary variation as functions of time, position and speed. It can also be coupled with external circuits. Rotational or translational motion effects can be included in the simulation.

Electrostatic Field Simulation

The electrostatic field simulator computes static electric fields arising from potential differences and charge distributions.

Electrostatic Theory

The electrostatic field simulator solves for the electric potential, $\phi(x,y)$, in this field equation:

$$\nabla\bullet(\varepsilon_r\varepsilon_0\nabla\phi(x,y)) = -\rho$$

where:

- $\phi(x,y)$ is the electric potential.
- ε_r is the relative permittivity. It can be different for each material.
- ε_o is the permittivity of free space, 8.854×10^{-12} F/m.
- $\rho(x,y)$ is the charge density.

This equation is derived from Gauss's Law and from Faraday's law of induction. Gauss's Law indicates that the net electric flux passing through any closed surface is equal to the net positive charge enclosed by that surface. In differential form, Gauss's Law is:

$$\nabla \bullet \boldsymbol{D} = \rho$$

where **D**(x,y) is the electric flux density. Since $D=\varepsilon_r\varepsilon_o E$, then:

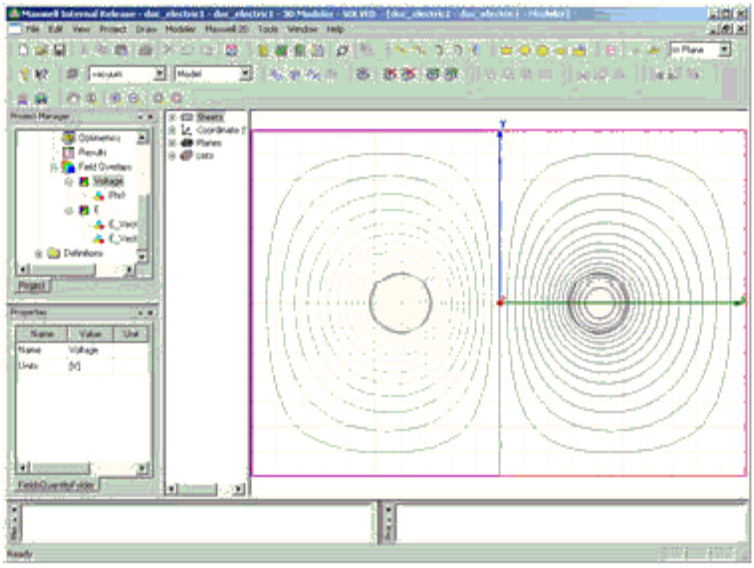
$$\nabla \bullet (\varepsilon_r \varepsilon_0 \boldsymbol{E}(x,y)) = \rho$$

In a static field, as a consequence of Faraday's law, $E = -\nabla\phi$. Therefore,

$$\nabla \bullet (\varepsilon_r \varepsilon_0 \nabla\phi(x,y)) = -\rho$$

which is the equation that the electrostatic field simulator solves using the finite element method. After the solution for the potential is generated, the system automatically computes the E-field and D-field using the relations $E = -\nabla\phi$ and $D=\varepsilon_r\varepsilon_o E$.

An contour plot of electric potential generated by the electrostatic field simulator is shown below:



Capacitance

At the simplest level, capacitance relates to the amount of energy stored in the electric field in and surrounding a structure. In a single circuit, the capacitance represents the amount of energy stored in the electric field that arises due to a voltage differential across a dielectric.

$$W_e = \frac{1}{2} C v^2$$

where W_e is the energy stored in the electric field, C is the capacitance, and v is the voltage across the dielectric.

Maxwell computes the capacitance between two conductors by simulating the electric field that arises when a voltage differential is applied. By computing the energy stored in the field, the corresponding capacitance can be computed.

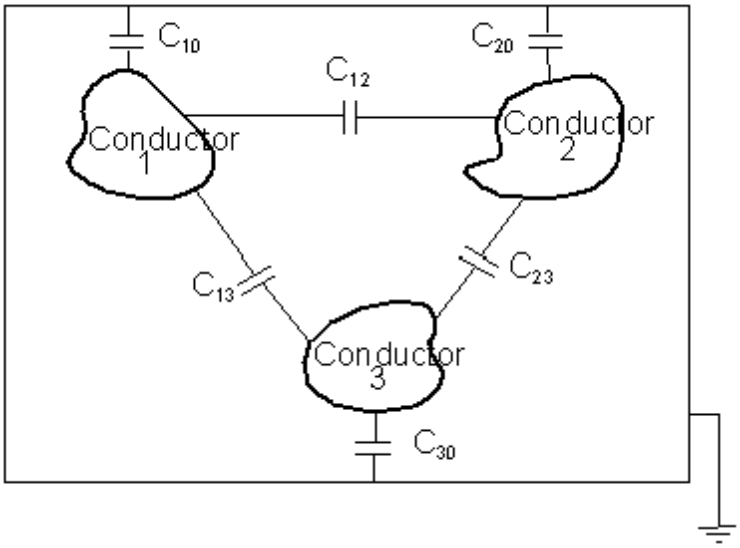
$$C = \frac{2W_e}{v^2}$$

To compute capacitances using this method, the E-field and D-field associated with a given distribution of voltages must first be computed. The electrostatic field simulator, which computes the electric potential at all points in the problem region, does this.

Capacitance in Terms of Charges and Voltages

A capacitance matrix represents the charge coupling within a group of conductors — that is, the relationship between charges and voltages for the conductors. Given the three conductors shown below, with the outside boundary taken as a reference, the net charge on each object will be:

$$Q_1 = C_{10} V_1 + C_{12}(V_1 - V_2) + C_{13}(V_1 - V_3)$$
$$Q_2 = C_{20} V_2 + C_{12}(V_2 - V_1) + C_{23}(V_2 - V_3)$$
$$Q_3 = C_{30} V_3 + C_{13}(V_3 - V_1) + C_{23}(V_3 - V_2)$$



This can be expressed in matrix form as:

$$\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} = \begin{bmatrix} C_{10}+C_{12}+C_{13} & -C_{12} & -C_{13} \\ -C_{12} & C_{20}+C_{12}+C_{23} & -C_{23} \\ -C_{13} & -C_{23} & C_{30}+C_{13}+C_{23} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

The capacitance matrix above gives the relationship between Q and V for the three conductors and ground. In a device with *n* conductors, this relationship would be expressed by an *n* x *n* capacitance matrix. Capacitance matrix values are specified in farads (coulombs/volt).

If one volt is applied to Conductor 1 and zero volts is applied to the other two conductors, the capacitance matrix becomes:

$$\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} = [C] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} C_{10}+C_{12}+C_{13} \\ -C_{12} \\ -C_{13} \end{bmatrix}$$

The diagonal elements in the matrix (such as $C_{(1,1)}$) are the sum of all capacitances from one conductor to all other conductors. These terms represent the self-capacitance of the conductors. Each is numerically equal to the charge on a conductor when one volt is applied to that conductor and the other conductors (including ground) are set to zero volts. For instance,

$C_{(1,1)} = C_{10} + C_{12} + C_{13}$

The off-diagonal terms in each column (such as $C_{(1,2)}$, $C_{(1,3)}$) are numerically equal to the charges induced on other conductors in the system when one volt is applied to that conductor. For instance, in column one of the example capacitance matrix, $C_{(1,2)}$ is equal to $-C_{12}$. This is equal to the charge induced on Conductor 2 when one volt is applied to Conductor 1 and zero volts are applied to Conductor 2.

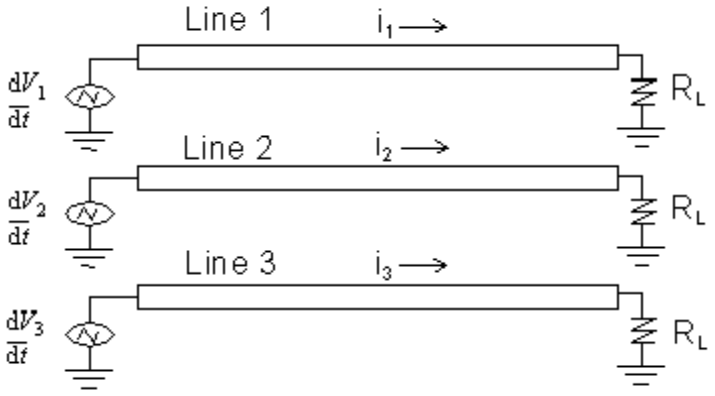
The off-diagonal terms are simply the negative values of the capacitances between the corresponding conductors (the mutual capacitances). In column one of the example capacitance matrix, the off-diagonal terms represent the capacitances between Conductor 1 and the other two conductors; in column two, the terms represent the capacitance between Conductor 2 and the other conductors; and so forth.

Note that the capacitance matrix is symmetric about the diagonal. This indicates that the mutual effects between any two objects are identical. For instance, $C_{(1,3)}$, the capacitance between Conductor 1 and Conductor 3 ($-C_{13}$), is equal to $C_{(3,1)}$, the capacitance between Conductor 3 and Conductor 1.

Capacitance in Terms of Currents and Time Varying Voltages

A capacitance matrix can also represent the relationship between currents and time varying voltages in a system of conductors.

Given the three transmission lines shown here,



the currents caused by the time varying voltage source on each line are given by the following relationship:

$$\begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = \begin{bmatrix} C_{10}+C_{12}+C_{13} & -C_{12} & -C_{13} \\ -C_{12} & C_{20}+C_{12}+C_{23} & -C_{23} \\ -C_{13} & -C_{23} & C_{30}+C_{13}+C_{23} \end{bmatrix} \begin{bmatrix} dV_1/(dt) \\ dV_2/(dt) \\ dV_3/(dt) \end{bmatrix}$$

If dV_2/dt and dV_3/dt are set to zero, this relationship becomes:

$$\begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = [C] \begin{bmatrix} dV_1/(dt) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} C_{10}+C_{12}+C_{13} \\ -C_{12} \\ -C_{13} \end{bmatrix} (dV_1/(dt))$$

This gives the currents that are induced on Line 2 and Line 3 when a time varying voltage source is applied to Line 1 — that is, the capacitive coupling between the three lines, or the short circuit capacitance.

Computing Capacitance

To compute a capacitance matrix for a structure, the Maxwell performs a sequence of electrostatic field simulations. In each field simulation, one volt is applied to a single conductor and zero volts is applied to all other conductors. Therefore, for an *n*-conductor system, *n* field simulations are automatically performed.

The energy stored in the electric field associated with the capacitance between two conductors is given by the following relation:

$$W_{ij} = \frac{1}{2}\int_{\Omega} \boldsymbol{D}_i \bullet \boldsymbol{E}_j d\Omega$$

where:

- *Wij* is the energy in the electric field associated with flux lines that connect charges on conductor *i* to those on conductor *j*.
- *Di* is the electric flux density associated with the case in which one volt is placed on conductor *i*.
- *Ej* is the electric field associated with the case in which one volt is placed on conductor *j*.

The capacitance between conductors *i* and *j* is therefore:

$$C = \frac{2W_{ij}}{v^2} = \int_{\Omega} \boldsymbol{D}_i \bullet \boldsymbol{E}_j d\Omega$$

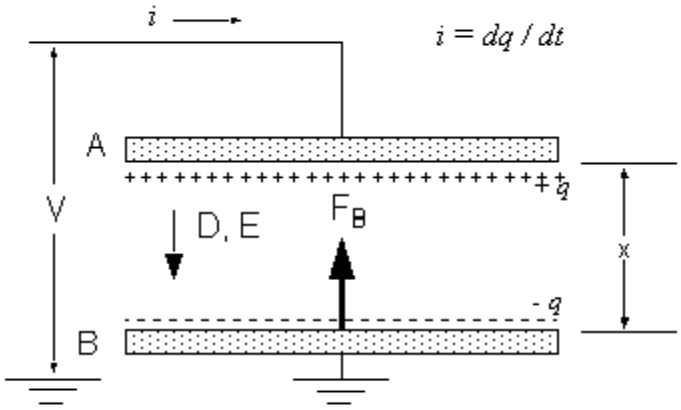
Virtual Forces (Electrostatic)

To compute the virtual force on an object, the electrostatic field simulator uses the principle of virtual work. In the structure shown below, the force on the bottom plate (plate B) in the direction of the displacement, x, is given by the following relationship:

$$F_B = \left.\frac{dW(v,x)}{dx}\right|_{V = Constant}$$

where W is the stored energy of the system,

$$W(v,x) = \frac{1}{2}\int_{Vol} \boldsymbol{E} \bullet \boldsymbol{D}\, dVol$$



Unlike the classical virtual work method, the plate is not actually moved during the numerical process of the force computation. Instead, only the triangles that lie along the outside surface of the object are virtually distorted. W and its derivative, dW/dx, are calculated from a single field solution using finite element interpolation functions.

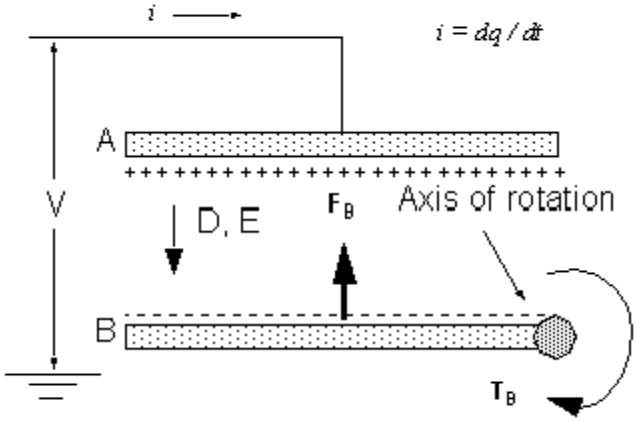
Virtual Torques (Electrostatic)

Similar to the virtual force calculation, the system uses virtual work principles to compute the torque on an object. In the structure shown below, the virtual torque on the bottom plate (plate B) about the axis of rotation is given by the following relationship:

$$T_B = \left.\frac{dW(v,\theta)}{d\theta}\right|_{V = Constant}$$

where W is the stored energy of the system,

$$W(v,\theta) = \frac{1}{2}\int_{Vol} \boldsymbol{E} \bullet \boldsymbol{D}\, dVol$$



Unlike the classical virtual work method, the plate is not actually rotated during the numerical process of the torque computation. Instead, only the triangles that lie along the outside surface of the object are virtually distorted. W and its derivative, dW/dθ, are calculated from a single field solution using finite element interpolation functions.

Flux Linkage (Electrostatic)

To compute the electric flux linkage, the electrostatic field solver uses the following relationship:

$$\Psi = \int \boldsymbol{E} \bullet d\boldsymbol{A}$$

where *E* is the electric field and *A* is the area over which flux density is computed.

- In cartesian (XY) models, the area is found by sweeping the flux line you've drawn in the xy-plane into the z direction — forming a 3D surface. The electric flux value computed is the

flux per meter depth in the z direction.

- In axisymmetric (RZ) models, the area is found by rotating the flux line you've drawn in the rz-plane 360 degrees about the z axis. The electric flux computed is the total flux that passes through this surface.

A separate flux linkage value is computed for each line you draw.

Magnetostatic Field Simulation

The magnetostatic field simulator lets you compute static magnetic fields arising from DC currents and other sources like permanent magnets and external magnetic fields. Magnetic fields in both linear and nonlinear materials can be simulated.

Magnetostatic Theory

The magnetostatic field simulator solves for the magnetic vector potential, $\mathbf{A}_Z(x,y)$ in this field equation:

$$\boldsymbol{J}_z(x,y) = \nabla\times\left(\frac{1}{\mu_r\mu_0}(\nabla\times\boldsymbol{A}_z(x,y))\right)$$

where:

- *Az(x,y)* is the z component of the magnetic vector potential.
- *Jz(x,y)* is the DC current density field flowing in the direction of transmission.
- μ_r is the relative permeability of each material.
- μ_0 is the permeability of free space.

Given *Jz(x,y)* as an excitation, the magnetostatic field simulator computes the magnetic vector potential at all points in space.

Note	In general, both **J** and **A** are vectors. However, **J** is assumed to only have a z-component. A consequence of this is that **A** only has a z-component as well. Both quantities can therefore be treated as scalars.

The equation that the magnetostatic field solver computes is derived from Ampere's law, which is:

$$\nabla\times\boldsymbol{H} = \boldsymbol{J}$$

$$\nabla\cdot\boldsymbol{B} = 0 \qquad H = \frac{B}{\mu_r\mu_0}$$

and from Maxwell's equation, . Since , then:

$$\nabla \times \left(\frac{\boldsymbol{B}}{\mu_r \mu_0} \right) = \boldsymbol{J}$$

$\boldsymbol{B} = \nabla \times \boldsymbol{A}$ $\nabla \cdot \boldsymbol{B} = 0$

Since , due to , then:

$$\nabla \times \left(\frac{1}{\mu_r \mu_0} \nabla \times \boldsymbol{A} \right) = \boldsymbol{J}$$

The magnetostatic field simulator solves this equation using the finite element method.

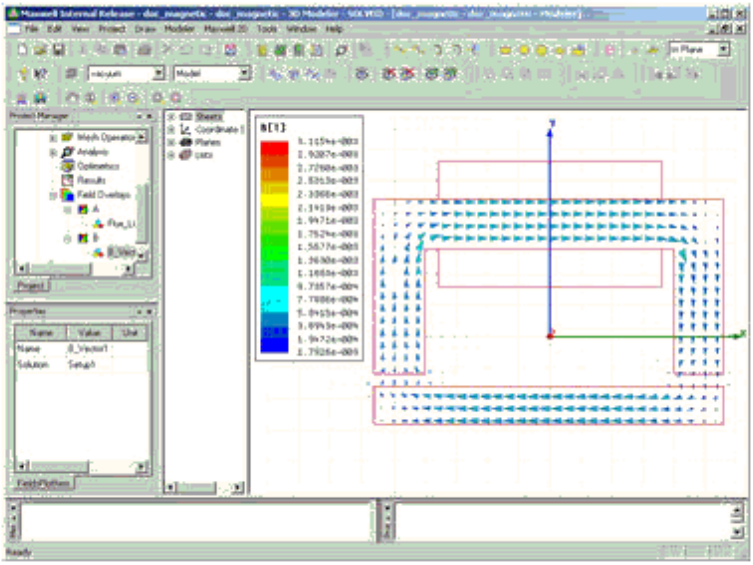
After $\mathbf{A}_Z(x,y)$ is computed, the magnetic flux density, **B**, and the magnetic field, **H**, can then be computed using the relationships:

$$\boldsymbol{B} = \nabla \times \boldsymbol{A}$$

$$\boldsymbol{H} = \frac{\boldsymbol{B}}{\mu_r \mu_0}$$

and **H**, for linear materials, is:

Both **B** and **H** lie in the xy cross-section being analyzed. An arrow plot of a B-field generated by the magnetostatic field simulator is shown below:



Inductance

At the simplest level, inductance relates to how much energy is stored in the magnetic field when current flows.

$$W_m = \frac{1}{2} L i^2$$

where:

- *Wm* is the energy stored in the magnetic field
- *L* is the inductance
- *i* is the current flowing in the circuit

The Maxwell computes inductances associated with a structure by simulating the magnetic field that arises when various voltages and currents are applied. Then, by computing the energy stored in those fields, it can then compute the necessary inductances:

$$L = \frac{2 W_m}{i^2}$$

To compute inductances using this method, the B-field and H-field associated with a distribution of currents must first be computed. The magnetostatic field simulator, which computes the magnetic vector potential at all points in the problem region, performs this task.

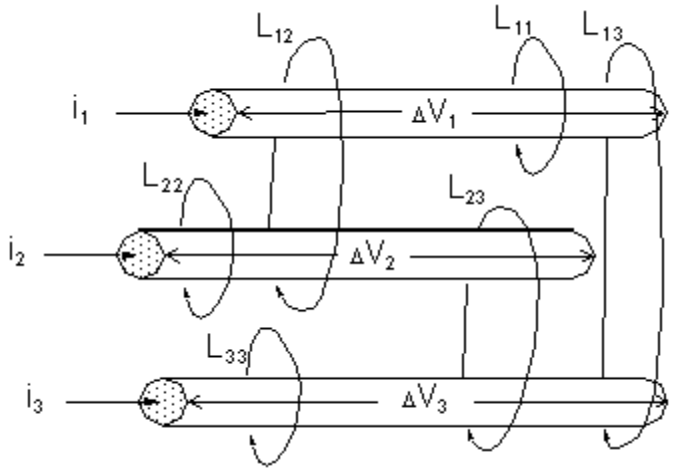
Inductance in Terms of Flux Linkage and Currents

An inductance matrix represents the magnetic flux linkage between the current loops in a system. Given the three current loops below, the relationship between induced flux and currents is as follows:

$\lambda_1 = L_{11}\, i_1 + L_{12}\, i_2 + L_{13}\, i_3$

$\lambda_2 = L_{12}\, i_1 + L_{22}\, i_2 + L_{23}\, i_3$

$\lambda_3 = L_{13}\, i_1 + L_{23}\, i_2 + L_{33}\, i_3$



This can be expressed in matrix form as:

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{12} & L_{22} & L_{23} \\ L_{13} & L_{23} & L_{33} \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix}$$

The inductance matrix above gives the relationship between λ and *i* for the three independent current loops. In a device with *n* current loops, this relationship would be expressed by an $n \times n$ inductance matrix. Inductance matrix values are specified in henries.

If one ampere is applied to Current Loop 1 and zero amperes are applied to the other two loops, the inductance matrix becomes:

$$\begin{bmatrix}\lambda_1\\ \lambda_2\\ \lambda_3\end{bmatrix} = \begin{bmatrix}L\end{bmatrix}\begin{bmatrix}1\\ 0\\ 0\end{bmatrix} = \begin{bmatrix}L_{11}\\ L_{12}\\ L_{13}\end{bmatrix}$$

The diagonal terms in the matrix (such as L_{11}) represent the self-inductance of each current loop. Self-inductance is numerically equal to the flux linkage in a current loop when one ampere is flowing in it, and no current is flowing in the other loops. For example, L_{11} is equal to the flux in Current Loop 1 when one ampere is flowing in that current loop, and no current is flowing in the other loops.

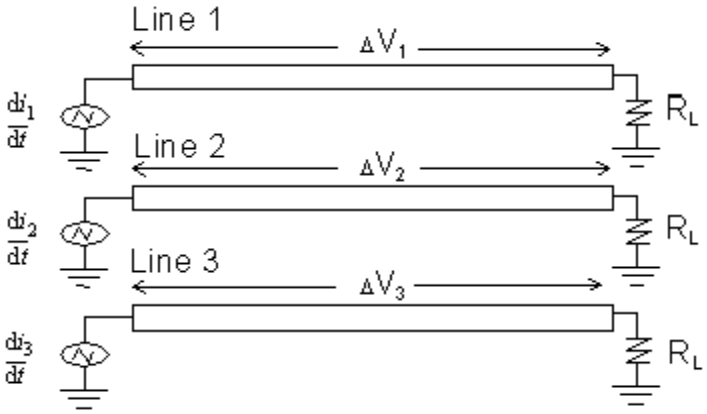
The off-diagonal terms (such as L_{12}, L_{13}) represent the mutual inductances between the current loops. Mutual inductance is numerically equal to the flux linkage in a current loop when one ampere is flowing through another loop, and no current is flowing anywhere else. For example, L_{12} is equal to the flux linkage in Loop1 when one ampere is applied to Loop 2 and no current is flowing in the other loops.

Note that the inductance matrix is symmetric about the diagonal. This indicates that the mutual effects between any two loops are identical. For instance, L_{13}, the inductance between Current Loop 1 and Current Loop 3, is equal to the inductance between Current Loop 3 and Current Loop 1.

Inductance in Terms of Voltages and Time Varying Currents

An inductance matrix can also represent the relationship between voltage and current fluctuations in a system. Given the three transmission lines shown below, the voltage changes caused by the time varying current source on each line are given by:

$$\begin{bmatrix}\Delta V_1\\ \Delta V_2\\ \Delta V_3\end{bmatrix} = \begin{bmatrix}L_{11} & L_{12} & L_{13}\\ L_{12} & L_{22} & L_{23}\\ L_{13} & L_{23} & L_{33}\end{bmatrix}\begin{bmatrix}(di_1)/(dt)\\ (di_2)/(dt)\\ (di_3)/(dt)\end{bmatrix}$$



The inductance matrix above gives the relationship between ΔV and *di/dt* for the three independent transmission lines.

If di_2/dt and di_3/dt are set to zero, this relationship becomes

$$\begin{bmatrix} \Delta V_1 \\ \Delta V_2 \\ \Delta V_3 \end{bmatrix} = [L] \begin{bmatrix} (di_1)/(dt) \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} L_{11} \\ L_{12} \\ L_{13} \end{bmatrix} (di_1)/(dt)$$

This gives the voltage changes that are induced on Lines 2 and 3 when a time-varying current source is applied to Line 1 — that is, the inductive coupling between all the loops.

Computing an Inductance Matrix

To compute an inductance matrix, the software performs a sequence of magnetostatic field simulations. In each field simulation, one ampere is allowed to flow in a single conductor. The current returns as defined under **Maxwell2D>Parameters>Assign>Matrix** — either in the conductor you identified as the return path, or along outside balloon, value (Dirichlet) or odd symmetry boundaries. No current flows in any other conductor.

For an n-conductor system, *n* field simulations are automatically performed. The energy stored in the magnetic field that couples two conductors is:

$$W_{ij} = \frac{1}{2} L I^2 = \frac{1}{2} \int_{\Omega} B_i \bullet H_j d\Omega$$

where:

- *Wij* is the energy stored in the magnetic field linking conductor *i* with conductor *j*.
- *I* is the current in conductor *i*.
- *Bi* is the magnetic flux density where one ampere is allowed to flow through conductor *i*.
- *Hj* is the magnetic field where one ampere is allowed to flow through conductor *j*.

The inductance coupling conductors *i* and *j* is therefore:

$$L_{ij} = \frac{2 W_{ij}}{I^2} = \int_{\Omega} B_i \bullet H_j d\Omega$$

For multiturn conductors, the net value of inductance is the value given by:

$$L_{net} = N^2 L_{matrix}$$

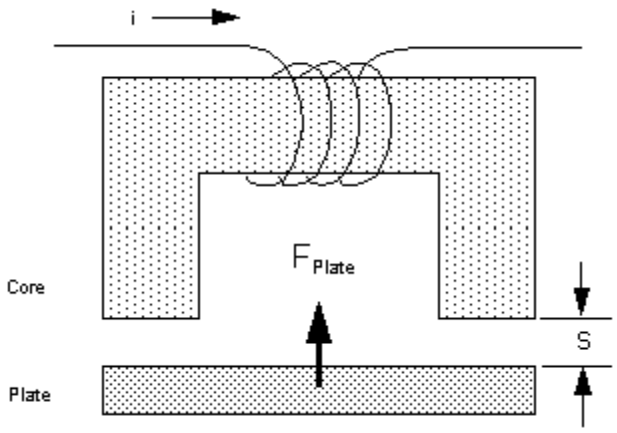
where *N* is the number of turns in the coil.

Virtual Forces (Magnetostatic)

To compute the force on an object, the system uses the principle of virtual work. In the structure shown below, the force on the plate in the direction of the displacement, *s*, is given by the following relationship:

$$\boldsymbol{F}_{plate} = \left. \frac{dW(s,i)}{ds} \right|_{i = const} = \frac{\partial}{\partial s} \left[\int_V \left(\int_0^H \boldsymbol{B} \bullet d\boldsymbol{H} \right) dV \right]$$

where W(s,i) is the magnetic coenergy of the system. The current, i, is held constant.



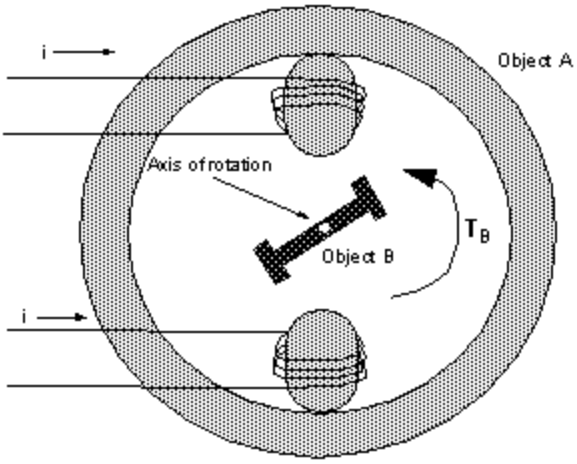
Unlike the classical virtual work method, the plate is not actually moved during the force computation. Instead, only the triangles that lie along the outside surface of the object are virtually distorted. Thus, the force computation only requires one field solution.

Virtual Torques (Magnetostatic)

Similar to the virtual force calculation, the system uses virtual work principles to compute the torque on an object. In the structure shown below, the torque on Object B about the axis of rotation is given by the following relationship:

$$T_B = \left.\frac{dW(\theta, i)}{d\theta}\right|_{i = const} = \frac{\partial}{\partial\theta}\left[\int_V \left(\int_0^H \boldsymbol{B} \bullet d\boldsymbol{H}\right) dV\right]$$

where W(θ, i) is the magnetic coenergy of the system. The current, i, is held constant.



Unlike the classical virtual work method, Object B is not actually rotated during the torque computation. Instead, only the triangles that lie along the outside surface of the object are virtually distorted. Thus, the change in the system's coenergy (and therefore the virtual torque) is given by the change in the coenergy of these triangles.

Flux Linkage (Magnetostatic)

To compute the magnetic flux linkage, the magnetostatic field solver uses the following relationship:

$$\Psi_i = L_{ii} I_i + \left(\sum_{j=1}^{n} L_{ij} I_j \right) + \Psi_m$$

where:

$L_{ii} I_i$ defines the flux linkage due to self-inductance in a coil,

$\sum_{j=1}^{n} L_{ij} I_j$ is the flux linkage due to mutual inductance between coils (i ≠ j), and

Ψ_m is the permanent magnetic flux linkage .

The energy method is used to calculate the inductance of a winding, which is then multiplied by the current flowing in the winding to get the flux linkage.

To obtain a correct flux vector, all current-carrying coils must be made part of the L-matrix computation. If you leave out L-entries that belong to current-carrying coils, the fluxes will be incorrect. (Refer to *Inductance in Terms of Flux Linkage and Currents* for additional information.)

Note	Maxwell no longer uses the following computation algorithm:

	$\psi = \int \boldsymbol{B} \bullet d\boldsymbol{A}$

Related Topics

Inductance in Terms of Flux Linkage and Currents

Magnetic Field Energy for a Magnetostatic Field Solution

Eddy Current Field Simulation

The eddy current field simulator allows you to simulate the effects of time-varying currents in parallel-conductor structures — including eddy current effects in conductors.

Related Topics

Eddy Current Theory

"Deriving the Eddy Current Equation" on page 28-23

"Eddy Currents and Skin Depth " on page 28-27

"Impedance Matrix " on page 28-27

"Virtual Forces (Eddy Current) " on page 28-32

"Virtual Torques (Eddy Current) " on page 28-34

"Current Flow (Eddy Current) " on page 28-34

"Nonlinear Eddy Current Field Simulation" on page 28-34

Eddy Current Theory

Time-varying currents flowing in a conductor produce a time-varying magnetic field in planes perpendicular to the conductor. In turn, this magnetic field induces eddy currents in the source conductor and in any other conductor parallel to it. The eddy current field solver calculates the eddy currents by solving for **A** and ϕ in the field equation:

$$\nabla \times \frac{1}{\mu}(\nabla \times \boldsymbol{A}) = (\sigma + j\omega\varepsilon)(-j\omega\boldsymbol{A} - \nabla\phi)$$

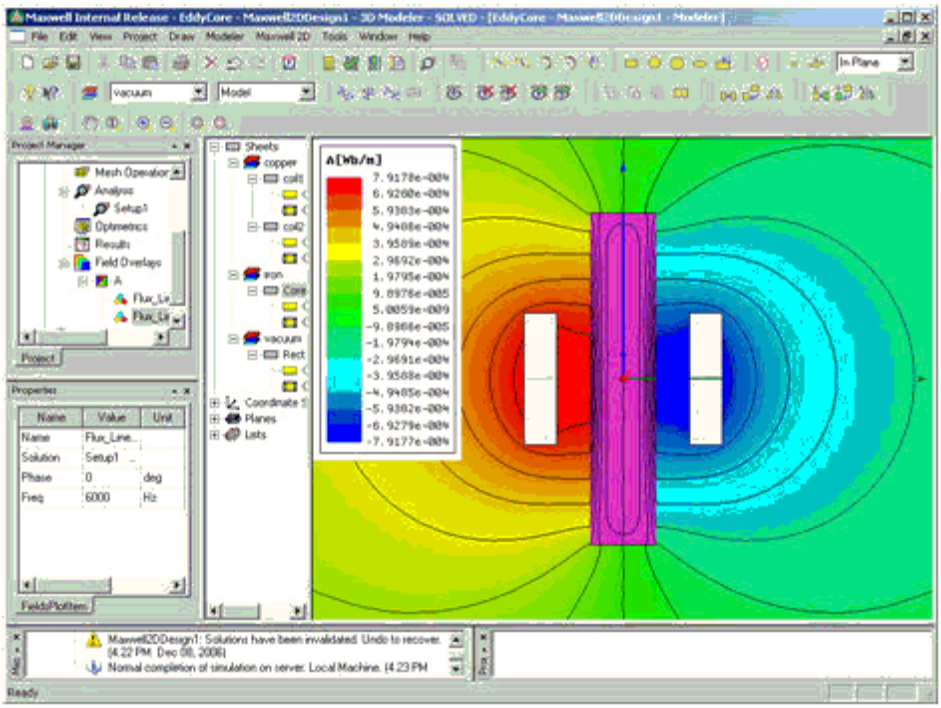
where:

- *A* is the magnetic vector potential.
- ϕ is the electric scalar potential.
- μ is the absolute magnetic permeability.
- ω is the angular frequency at which all quantities are oscillating.

- σ is the conductivity.
- ε is the absolute permittivity.

Note	The eddy current equation is derived from Maxwell's equations. Phasor notation is used to represent complex quantities.

A plot of flux lines produced by eddy currents that were computed in a structure by the eddy current solver is shown below:



Related Topics

"Components of Current Density " below
"Integrating the Current Density " on the next page
"Eddy Current Assumptions " on the next page

Components of Current Density

Notice that the right side of the equation:

$$\nabla \times \frac{1}{\mu}(\nabla \times \boldsymbol{A}) = (\sigma + j\omega\varepsilon)(-j\omega\boldsymbol{A} - \nabla\phi)$$

$(-j\omega \boldsymbol{A} - \nabla\phi)$

consists of a complex conductivity multiplied by the complex value of **E** . It is therefore equal to the complex current density, **J**, which has three components:

- $-\sigma\nabla\phi$
  $\mathbf{J_s}$, the source current density due to differences in electric potential, .
- $\mathbf{J}_e$, the induced eddy current density due to time-varying magnetic fields, - $j\omega\sigma\mathbf{A}$.
- $j\omega\varepsilon(-j\omega \boldsymbol{A} - \nabla\phi)$
  $\mathbf{J}_d$, the displacement current density (time-varying electric fields), .

The total current density is the sum of these three components. The $j\omega$ term in the eddy and displacement components indicate that they are a function of frequency and become increasingly significant as the frequency increases.

Integrating the Current Density

When setting up a problem, you specify the total current flowing in any conductor that is connected to an external source. Therefore, the eddy current module is able to make use of a second equation:

$$I_T = \int_\Omega (\sigma + j\omega\varepsilon)(-j\omega \boldsymbol{A} - \nabla\phi) d\Omega$$

which simply reflects the fact the total current in a conductor equals the integral of the current density over the cross-section of the conductor, Ω.

Eddy Current Assumptions

The eddy current field solver makes the following assumptions about the field quantities for which it solves:

- Time-varying electromagnetic quantities are assumed to have the periodic waveform:

$$\boldsymbol{F}(t) = F_m \cos(\omega t + \theta)$$

- All quantities must have the same frequency (angular frequency), but can have different phase angles (θ). If a current is not a pure sinusoid, decompose it into sinusoidal harmonics, and solve separately at each frequency, but only for linear problems.
- All currents (source, eddy, and displacement) are assumed to flow perpendicular to the plane being studied (that is, in the z-direction for XY applications). Therefore, the magnetic

fields associated with these currents lie within the xy-plane. As a result, **A**, the magnetic vector potential, has a z-component only.

- Because no currents flow in the xy-plane, the electric field, **E**, has a z-component only. It follows that ϕ is constant over the cross-section of each conductor in the problem.

Deriving the Eddy Current Equation

The eddy current field solver uses the finite element method to compute **A** and ϕ using these two relationships:

$$\nabla \times \frac{1}{\mu}(\nabla \times \boldsymbol{A}) = (\sigma + j\omega\varepsilon)(-j\omega\boldsymbol{A} - \nabla\phi)$$

$$I_T = \int_{\Omega} (\sigma + j\omega\varepsilon)(-j\omega\boldsymbol{A} - \nabla\phi)d\Omega$$

where:

- A is the magnetic vector potential.
- ϕ is the electric scalar potential.
- μ is the magnetic permeability.
- ω is the angular frequency at which all quantities are oscillating.
- σ is the conductivity.
- ε is the permittivity.
- I_T is the total current flowing in conductors.

The following section shows how these equations are derived from Maxwell's equations.

Maxwell's Equations

The eddy current field simulator solves for time harmonic electromagnetic fields governed by Maxwell's equations:

$$\nabla \times \boldsymbol{H} = \boldsymbol{J} + \frac{\partial \boldsymbol{D}}{\partial t}$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}$$

$$\nabla \bullet \boldsymbol{D} = \rho$$

$$\nabla \bullet \boldsymbol{B} = 0$$

where:

- $\boldsymbol{E}$ is the electric field.
- $\boldsymbol{D}$ is the electric displacement, $\varepsilon \boldsymbol{E}$.
- $\boldsymbol{B}$ is the magnetic flux density.
- $\boldsymbol{H}$ is the magnetic field intensity, $\mathbf{B}/\mu$.
- $\boldsymbol{J}$ is the conduction current density, $\sigma \boldsymbol{E}$.
- ρ is the charge density.

The eddy current solver assumes that all time-varying electromagnetic quantities in the problem have the form:

$$\boldsymbol{F}(t) = F_m \cos(\omega t + \theta)$$

Using Euler's formula:

$$e^{j\alpha} = \cos\alpha + j\sin\alpha$$

If $\alpha = \omega t + \theta$, $\mathbf{F}(t)$ equals the real portion of $e^{j(\omega t+\theta)}$:

$$\boldsymbol{F}(t) = \Re\left[F_m e^{j(\omega t+\theta)}\right] = \Re[F_m(\cos(\omega t+\theta) + j\sin(\omega t+\theta))] = F_m \cos(\omega t+\theta)$$

Now, because each time-varying quantity has the form $F_m e^{j\theta} e^{j\omega t}$, $\frac{\partial \boldsymbol{D}}{\partial t}$ and $\frac{\partial \boldsymbol{B}}{\partial t}$ are equal to $j\omega\mathbf{D}$ and $j\omega\mathbf{B}$.

Therefore, with this simplification and the relations $\mathbf{H}=\mu\mathbf{B}$, $\mathbf{D}=\varepsilon\mathbf{E}$, and $\mathbf{J}=\sigma\mathbf{E}$, Maxwell's equations reduce to:

$$\nabla\times\frac{1}{\mu}\boldsymbol{B} = (\sigma\boldsymbol{E} + j\omega\varepsilon\boldsymbol{E})$$

$$\nabla\times\boldsymbol{E} = -j\omega\boldsymbol{B}$$

$$\nabla\bullet\varepsilon\boldsymbol{E} = \rho$$

$$\nabla\bullet\boldsymbol{B} = 0$$

Relationship of Magnetic and Electric Field

The quantity that the eddy current field simulator actually solves for is **A**, the magnetic vector potential. It is given by:

$$\nabla\times\boldsymbol{A} = \boldsymbol{B}$$

Substituting this into the first of Maxwell's equations, the result is:

$$\nabla\times\frac{1}{\mu}(\nabla\times\boldsymbol{A}) = (\sigma\boldsymbol{E} + j\omega\varepsilon\boldsymbol{E})$$

A solution for **E** in terms of **A** is given by:

$$\boldsymbol{E} = -j\omega\boldsymbol{A} - \nabla\phi$$

where ϕ is the electric potential. Substituting the right side of this relationship for **E** into the previous equation results in:

$$\nabla\times\frac{1}{\mu}(\nabla\times\boldsymbol{A}) = (-j\omega\boldsymbol{A} - \nabla\phi)(\sigma + j\omega\varepsilon)$$

This equation is one of the two used by the eddy current solver to compute **A** and ϕ.

Relationship of Current and Current Density

Notice that the previous equation is in the form of a complex conductivity, $(\sigma+j\omega\varepsilon)$, times the complex value of **E**. The result is the complex current density, **J**. Therefore, the integral of this expression over the cross-section of a conductor is constrained to equal the total current that is specified as flowing in that conductor:

$$I_t = \int_\Omega \boldsymbol{J}d\Omega = \int_\omega \frac{1}{\mu}(\sigma + j\omega\varepsilon)(-j\omega\boldsymbol{A} - \nabla\phi)d\omega$$

This is the second equation that the eddy current solver uses in computing **A** and ϕ.

The total current, *It*, in this equation is the quantity you specify when setting up a problem. It is the total current flowing in a conductor, and includes:

- $-\int_\Omega \sigma\nabla\phi d\Omega$

  I_s, the current from an external source, .

- $-\int_\Omega j\omega\sigma\boldsymbol{A}d\Omega$

  I_e, the induced eddy current, .

- $\int_\Omega j\omega\varepsilon(-j\omega\boldsymbol{A} - \nabla\phi)d\Omega$

  I_d, the displacement current, .

For problems solved by the eddy current solver, **B** is assumed to lie in the xy plane. Therefore, **A** can only have a component in the z-direction. The simulator does not have to solve for the x and y components of **A**. Since **E** only has a z-component, ϕ is a constant for each cross-section of a conductor. Therefore, Maxwell does not have to solve for ϕ at every node.

For a more detailed discussion of eddy current and skin effect problems in multiconductor systems, see:

J. Weiss, Z. Cendes. "A One-Step Finite Element Method for Multiconductor Skin Effect Problems", *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-101, No. 10 October 1982.

Eddy Currents and Skin Depth

Induced currents allow magnetic fields to penetrate conductors only to a certain depth, which is approximated by the formula:

$$\delta = \sqrt{\frac{2}{\omega \sigma \mu_0 \mu_r}} \quad \text{(in meters)}$$

where:

- ω is the angular frequency, which is equal to $2\pi f$. (f is the frequency at which source currents and voltages oscillate during the solution.)
- σ is the conductor's conductivity, in siemens/meter.
- μ_r is the conductor's relative permeability, in amperes/meter.
- μ_0 is the permeability of free space, which is equal to $4\pi \times 10^{-7}$ A/m.

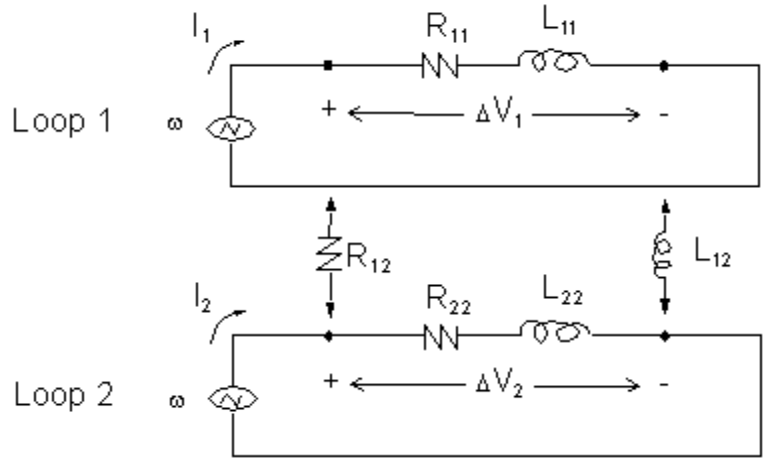
Currents will be concentrated near the surface of the conductor, decaying rapidly past the skin depth. As the formula above indicates, the skin depth gets smaller as the frequency increases.

Impedance Matrix

An impedance matrix summarizes the relationship between AC voltages and AC currents in multi-conductor systems. Given the two current loops below, the relationships between voltages and currents in each loop is as follows:

$$(\Delta V_1 = I_1 R_{11} + I_2 R_{12} + I_1 j\omega L_{11} + I_2 j\omega L_{12}) Size$$

$$\Delta V_2 = I_2 R_{22} + I_1 R_{12} + I_2 j\omega L_{22} + I_1 j\omega L_{12}$$



This can be expressed in matrix form as:

$$\begin{bmatrix} \Delta V_1 \\ \Delta V_2 \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{12} & Z_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

where:

- *Vi* and *Ii* are phasors.
- *Z11* = $R_{11} + j\omega L_{11}$ (the self-impedance of Loop 1).
- *Z12* = $R_{12} + j\omega L_{12}$ (the mutual impedance between Loops 1 and 2).
- *Z22* = $R_{22} + j\omega L_{22}$ (the self-impedance of Loop 2).

The impedance matrix above gives the relationship between V and I for the two current loops. In a device with *n* current loops, this relationship would be expressed by an *n* x *n* impedance matrix. The matrix values displayed by the software are resistance and inductance (not resistance and reactance) and therefore do not include jω.

Note	All impedances are complex numbers in the form: $$Z = R + j\omega L$$ where:

	• ω is equal to $2\pi f$, where f is the frequency of the AC current source. • R is the AC resistance, given in ohms/meter (XY) or ohms (RZ). • L is the AC inductance, given in henries/meter (XY) or henries (RZ).

Related Topics

"Computing an Impedance Matrix " below
"Virtual Forces (Eddy Current) " on page 28-32
"Virtual Torques (Eddy Current) " on page 28-34
"Current Flow (Eddy Current) " on page 28-34

Computing an Impedance Matrix

Maxwell breaks down the impedance matrix computation into two parts. First, it solves for the inductance matrix (L-matrix) associated with the model. It then solves for the resistance matrix (R-matrix). When it finishes solving for these matrices, the simulator combines them to form the impedance matrix, using the relationship $Z=R+j\omega L$.

To compute the inductance and resistance matrices for the impedance solution, the simulator generates an eddy-current field solution for each conductor in the matrix.

In the first solution, the current in the first conductor is set to one ampere; currents in the other conductors that are included in the impedance matrix are set to zero. This is done by imposing current sources on the conductors.

In the second solution, the current in the second conductor is set to one ampere, and all other conductors that are included in the impedance matrix are set to zero amperes, and so forth. Conductors that are not included in the impedance matrix are not affected.

Related Topics

"Matrix Inductance " below
"Matrix Resistance " on page 28-31
"Inductance and Resistance in Impedance Computations " on page 28-31

Matrix Inductance

To compute the inductance of the current loop, the simulator calculates the average energy, W_{AV}, of the system after a field solution is computed:

$$W_{AV} = \frac{1}{4}\int_V \boldsymbol{B} \bullet \boldsymbol{H}^* dV$$

Since the instantaneous energy of the system is equal to:

$$W_{Inst} = \frac{1}{2} L i^2$$

where the instantaneous value of the current is related to the peak value of the current by $i = I_{Peak}\cos(\omega t+\theta)$. The average value for the energy can then be found by integrating the instantaneous energy:

$$W_{AV} = \frac{1}{2\pi}\int_0^{2\pi} W_{Inst} d\omega t = \left(\frac{L}{2}\right)\left(\frac{1}{2\pi}\right)\int_0^{2\pi} I_{Peak}^2 [\cos(\omega t + \theta)]^2 d\omega t$$

From this, the average energy of the system is equal to:

$$W_{AV} = \left(\frac{L}{2}\right) I_{RMS}^2 = \left(\frac{L}{2}\right)\left(\frac{I_{Peak}}{\sqrt{2}}\right)^2 = \left(\frac{L}{4}\right) I_{Peak}^2$$

The inductance, therefore is:

$$L = \frac{4W_{AV}}{I_{Peak}^2}$$

The software assumes that the object for which impedance is being computed has a peak current of one ampere per coil turn flowing through it. Thus, the inductance is simply $4W_{AV}$.

The eddy current simulator includes the eddy effect in conductors when it computes inductance during an impedance solution.

Matrix Resistance

To compute the resistance, the simulator calculates the ohmic loss, P, after a field solution has been computed:

$$P = \frac{1}{2\sigma}\int_V \boldsymbol{J} \bullet \boldsymbol{J}^* dV$$

The ohmic loss is related to the resistance by:

$$P = RI_{RMS}^2$$

The resistance is therefore:

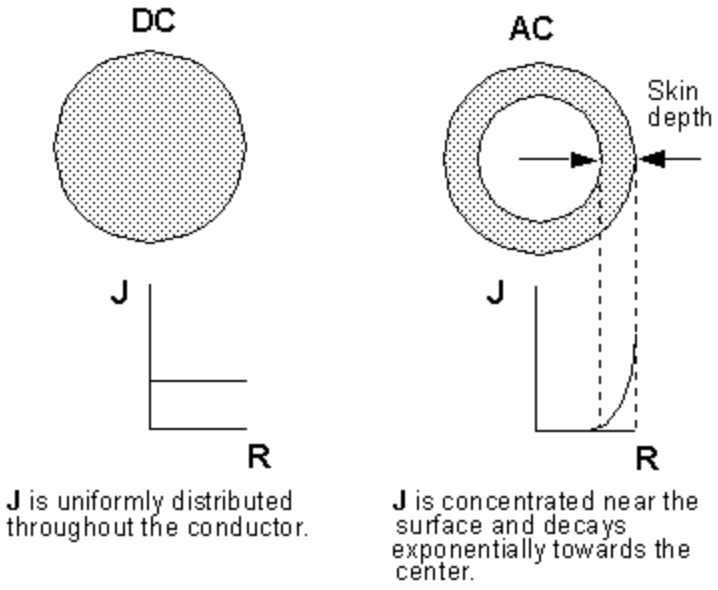
$$R = \frac{P}{I_{RMS}^2} = \frac{2P}{I_{Peak}^2}$$

The system assumes that the object for which impedance is being computed has a peak current of one ampere per coil turn flowing through it. Therefore, the resistance is simply 2P.

Note that the resistance for an eddy current problem will be higher than the equivalent DC resistance, due to the skin concentration of currents.

Inductance and Resistance in Impedance Computations

The inductances and resistances computed during an impedance matrix solution are different from those computed for the equivalent DC case. This figure shows how they differ:



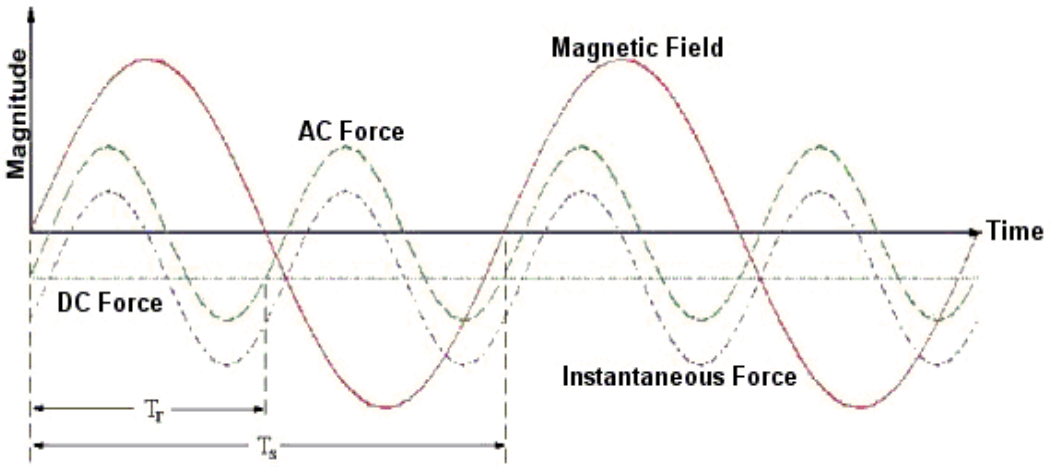
The current density, **J**, in the DC case (the conductor on the left) is evenly distributed throughout the cross-section of the conductor. The current density in the AC case (the conductor on the right) is distributed non-uniformly on the surface due to skin effect. Since the area through which current can flow is smaller, it follows that the resistance to the current flow is higher in the impedance matrix than in a resistance matrix computed for the equivalent DC case.

In the DC example, no eddy currents occur. The magnetic field created by the current flowing through the conductor is static. In the AC example, the oscillating magnetic field induces currents in conductors in the model. These induced currents affect the computation of inductance for the impedance matrix, causing it to be different from the equivalent DC computation of inductance.

Virtual Forces (Eddy Current)

Virtual force in an eddy current problem is computed the same way as virtual force in a magnetostatic problem. The only difference is that the average value of force over time is computed — not the instantaneous force at a given time.

The difference between the time-averaged (or DC) force, AC force, and instantaneous force is shown below:



Force oscillates at twice the frequency of the source current and magnetic field:

$$f_F = \frac{1}{T_F} = 2f_S$$

where:

- f_F is the frequency of the force.
- f_S is the frequency of the source current and magnetic field.
- T_F is the period of the force.

The time-averaged (or DC) force, AC force, and instantaneous force can be determined by:

$$F_{DC} = \frac{1}{2}\int Re\left|\bar{J} \times \bar{B}^{*}\right| dV$$

$$F_{AC} = \frac{1}{2}\int Re\left|\bar{J} \times \bar{B}\right| dV$$

$$F_{INST} = F_{DC} + F_{AC}$$

The AC force, F_{AC} must be evaluated at a particular phase (=ωt) in order to determine its magnitude at an instant in time. However, the peak value of the AC force is reported as the “AC Magnitude” in the force and torque solution panel for Maxwell 2D.

Virtual Torques (Eddy Current)

Virtual torque in an eddy current problem is computed the same way as virtual torque in a magnetostatic problem. The only difference is that the average value of the torque over time is computed, not the net torque at a given time.

Current Flow (Eddy Current)

To compute the current flow, the eddy current field solver uses the following relationship:

$$I = \int \boldsymbol{J} \bullet d\boldsymbol{A}$$

where:

- *I* is the current.
- ***J*** is the current density, given by:

$$\boldsymbol{J} = (\sigma + j\omega\varepsilon_r)(-j\omega\boldsymbol{A} - \nabla\phi)$$

- *A* is the area over which the current flow is computed.
  - In cartesian (XY) models, the area is found by sweeping the current flow line you’ve drawn in the xy-plane into the z direction — forming a 3D surface. The current flow computed is the current per meter depth in the z-direction.
  - In axisymmetric (RZ) models, the area is found by revolving the flux line you’ve drawn in the rz-plane 360 degrees around the z-axis, forming a 3D surface. The current flow computed is the total current that passes through this surface.

A separate current flow value is computed for each line you draw.

Nonlinear Eddy Current Field Simulation

Note	Applicable to both Maxwell 2D and 3D Eddy current solvers.

The Maxwell 2D and 3D Eddy current solvers, for designs with nonlinear materials, allow you to analyze the fundamental components of B and H at a specified frequency.

Related Topics

"Nonlinear Eddy Current Theory " below

"Sinusoidal B " below

Nonlinear Eddy Current Theory

Note	Applicable to both Maxwell 2D and 3D Eddy current solvers.

Unlike linear problems in which the field solution is assumed to be based on the sinusoidal complex peak field values in a steady state, nonlinear problems are usually based on all harmonic components of the fields.

That is to say for nonlinear problems, though the current is sinusoidal at its input, the resulting fields are “harmonically rich.” Although the B- and H-fields are not sinusoidal at output, the Maxwell 2D and 3D non-linear Eddy current solvers use equivalent fundamental components of B and H to approximate non-linear field behavior associated with the assigned non-linear BH characteristic.

Related Topics

"Sinusoidal B " below

Sinusoidal B

Note	Applicable to both Maxwell 2D and 3D Eddy current solvers.

Assuming both that **B** is sinusoidal with time, **t**, and that the value of **H** is derived from the original BH-curve, introduce a new value $\mathbf{H_e}$ to represent the effective magnetic field.

The derivation of the effective H_e is based on the equivalence of averaged energy between using original, non-sinusoidal H field and the effective sinusoidal field $H_e\sin(\omega t)$. That is:

$$\frac{4}{T}\int_0^{\frac{T}{4}} H_e \sin(\omega t) B_m \sin(\omega t) dt = \frac{4}{T}\int_0^{\frac{T}{4}} (H \cdot B) dt$$

with:

$$T = \frac{2\pi}{\omega}$$

the above expression becomes:

$$\int_0^{\frac{\pi}{2}} H_e B_m \sin^2(\omega t)\ d\omega t = \int_0^{\frac{\pi}{2}} (H \cdot B) d\omega$$

Allowing $\alpha=\omega t$ and extracting $\mathbf{H_e}$, simplifying yields:

$$H_e B_m \int_0^{\frac{\pi}{2}} \left(\frac{1 - \cos 2\alpha}{2}\right) d\alpha = \int_0^{\frac{\pi}{2}} (H \cdot B) d\alpha$$

or:

$$H_e = \frac{4}{\pi} \int_0^{\frac{\pi}{2}} \frac{(H \cdot B)}{B_m} d\alpha$$

where:

- *Bm* corresponds to the last point on the original BH characteristic (in the saturation portion) and which is made equal with $\pi/2$.

The above equation is used to derive an effective (B_m, H_e) characteristic for the frequency domain simulation. This process is required to be performed only once for a particular BH curve. During the simulation, H_e is determined for a specific B_m such that an equivalent permeability can be computed as the ratio B_m / H_e and used in the calculations.

DC Conduction Field Simulation

The DC conduction field simulator allows you to analyze conduction currents due to static electric fields in conductors and lossy dielectrics.

DC Conduction Theory

When a material with a non-zero conductivity is subject to a potential difference, conduction current flows in the material. At all points in the problem space, the current density (**J**) will be proportional to the electric field (**E**) that is established due to the potential difference.

$$\boldsymbol{J}(x,y) = \sigma \boldsymbol{E}(x,y) = -\sigma \nabla \phi(x,y)$$

where:

- *J(x,y)* is the current density.
- *E(x,y)* is the electric field.
- σ is the conductivity of the material in MKS units (S/m).
- ϕ*(x,y)* is the electric potential.

The equation that the DC conduction field simulator solves is based on the fact that, under steady state conditions, the amount of charge, ρ, leaving any infinitesimally small region must equal the charge flowing into that region.

$$\nabla \bullet \boldsymbol{J} = -\frac{\partial \rho}{\partial t} = 0$$

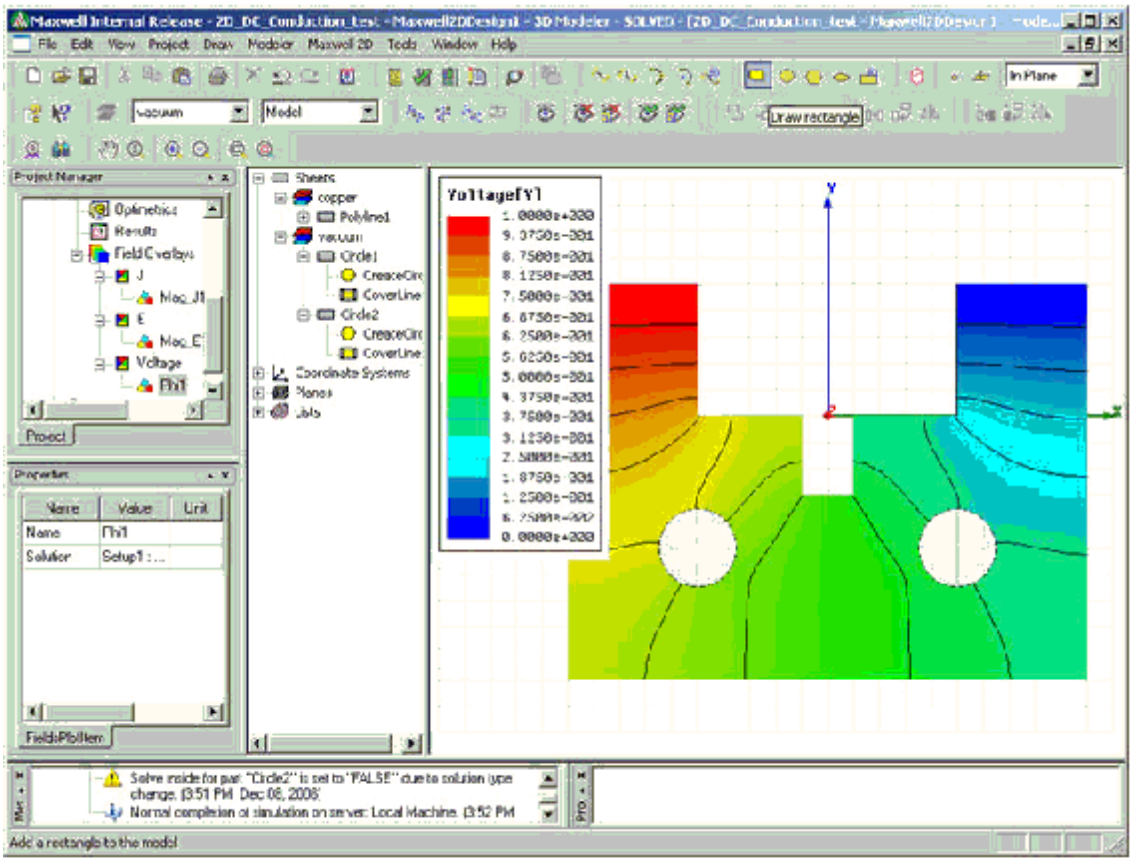
The field quantity that DC conduction actually solves for is the electric potential, ϕ, in the following equation:

$$\nabla \bullet (\sigma \nabla \phi) = 0$$

$$-\sigma \nabla \phi = \boldsymbol{J}$$

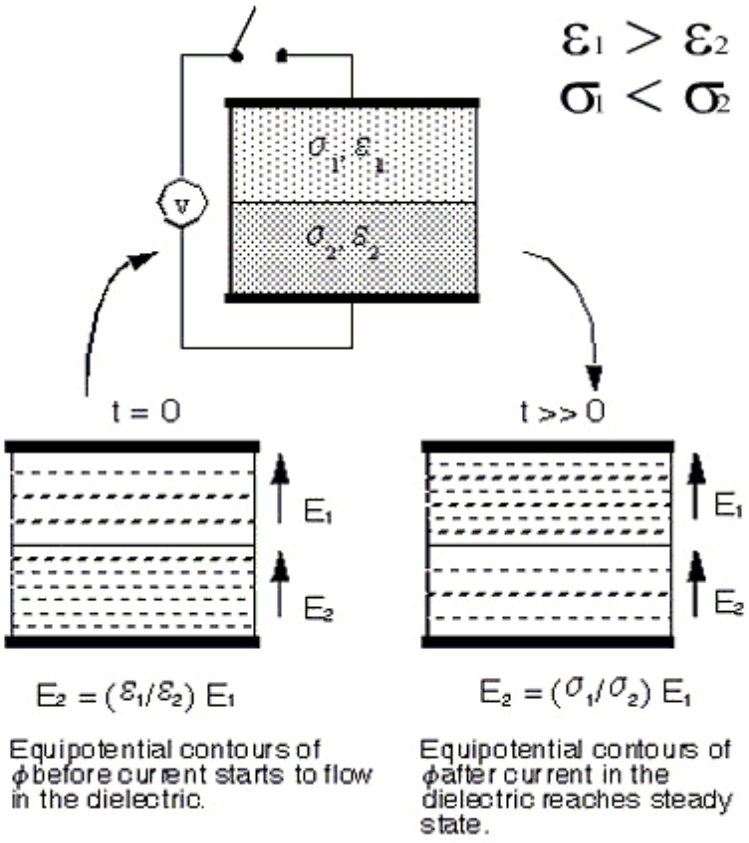
Note that .

A plot of electric potential that was computed by the DC conduction solver is shown below:



Steady-state Conditions

The DC conduction solver assumes that current flow in the conducting material has already reached steady state conditions. The implication of this assumption becomes clear when you compare the initial and steady state conditions of a simple problem such as that shown here:



First, consider the interval before steady state is reached. Assume that the switch establishing the potential across the parallel plates in the figure above closes at t=0. Also assume that the current required to deposit charges on the parallel plates (so that the voltage difference can be supported) occurs instantaneously.

However, it will take a while for current to start flowing in the dielectric. The time it takes for current to flow is determined by the time constant of the material. Therefore, at t=0, there will be an electric field in the dielectric, but no current and no free charges. Consequently, the relationship that must be satisfied at the interface of the two dielectrics is:

$$\nabla \bullet \boldsymbol{D} = \rho$$

or, equivalently:

$$\nabla \bullet \varepsilon \boldsymbol{E} = \rho$$

At the interface between the two dielectrics, this relationship implies that:

$$\varepsilon_1 \boldsymbol{E}_{n1} = \varepsilon_2 \boldsymbol{E}_{n2} \Rightarrow \boldsymbol{E}_{n1} = \frac{\varepsilon_2}{\varepsilon_1} \boldsymbol{E}_{n2}$$

where $\mathbf{E}_n$ is the normal component of **E**. Therefore, at t=0, before current starts to flow in the two dielectrics, $\phi(x,y)$ is determined entirely by the permittivity of the dielectrics. Use the electrostatic field solver to solve for $\phi(x,y)$ in such a case.

After current starts to flow in the dielectric and steady state is reached, free charges are able to accumulate at the boundaries of the two dielectrics. The free charge, ρ, is no longer zero.

$$\nabla \bullet \boldsymbol{J} = -\frac{\partial \rho}{\partial t} = 0$$

The relationship that must be satisfied now is:

$$\nabla \bullet \sigma \boldsymbol{E} = 0$$

or, equivalently,

This relationship implies that:

$$\sigma_1 \boldsymbol{E}_{n1} = \sigma_2 \boldsymbol{E}_{n2} \Rightarrow \boldsymbol{E}_{n1} = \frac{\sigma_2}{\sigma_1} \boldsymbol{E}_{n2}$$

In this case, the solution depends on the conductivity (σ) of the materials rather than on their dielectric constant (ε). The DC conduction solver analyzes the steady state condition. Therefore, use the DC conduction field solver when steady state conditions have been reached.

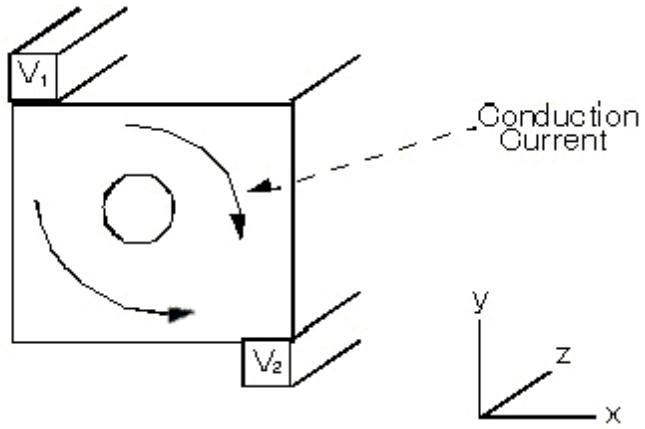
Relevant Time Constant

The time constant that determines how long it will take the current flowing in the conductive material to reach steady state is given by $\tau=\varepsilon/\sigma$. In a good conductor such as steel ($\varepsilon=9\text{x}10^{-12}$, $\sigma=1\text{x}10^{7}$), $\tau = 9\text{x}10^{-19}$ seconds. Steady state conditions are reached almost instantaneously and in such cases you should use the DC conduction solver to analyze the potential field.

On the other hand, for a good dielectric such as fused quartz ($\sigma=1\text{x}10^{-17}$), the time constant is on the order of 10^5 seconds. There will be a significant period before steady state currents start to flow and you should use the electrostatic solver to analyze the electric field. For many materials in between, a complete analysis may require an electric transient field solution.

Conductance

A conductance matrix gives the relationship between currents and voltage drops. In the following figure, two conductors at voltages V_1 and V_2 are touching a bar with a conductance, G.



Because the bar is not a perfect insulator, current will flow between the two conductors. Given the conductors shown on the previous page, the relationship between the conduction current and the voltage drop, V, in each conductor is:

$$I = GV$$

where:

- *I* is the conduction current.
- *G* is the conductance, measured in Siemens.
- *V* is the voltage drop given by V_1-V_2.

Conductance, as evident from the equation, is the inverse of resistance. Therefore, if a material has a large conductance, it will be a better conductor, and if it has a low conductance, it will be a better resistor.

Current Flow (DC Conduction)

To compute current flow, the DC conduction field solver uses the following relationship:

$$I = \int_A \boldsymbol{J} \bullet d\boldsymbol{A}$$

where:

- *I* is the current.
- *J* is the current density, given by:

$$\boldsymbol{J} = \sigma \boldsymbol{E} = -\sigma \nabla \phi$$

- *A* is the area over which the current flow is computed.
  - In cartesian (XY) models, the area is found by sweeping the current flow line you've drawn in the xy-plane into the z direction — forming a 3D surface. The current flow computed is the current per meter depth in the z direction.
  - In axisymmetric (RZ) models, the area is found by revolving the flux line you've drawn in the rz-plane 360 degrees around the z-axis, forming a 3D surface. The current flow computed is the total current that passes through this surface.

A separate current flow value can be computed for each line you draw.

AC Conduction Field Simulation

The AC conduction field solver allows you to analyze conduction currents caused by time-varying electric fields in conductors and lossy dielectrics.

AC Conduction Theory

The AC conduction field simulator solves for ϕ in the following equation:

$$\nabla \bullet [\sigma \boldsymbol{E} + j\omega\varepsilon \nabla \phi(x, y)] = 0$$

where:

- $\phi(x,y)$ is the magnitude and phase of the electric potential at each value of x and y.
- ω is the angular frequency at which the potential is oscillating.
- σ is the conductivity.
- ε is the permittivity.

The above equation is derived from:

$$\nabla \bullet (\boldsymbol{J} + j\omega \boldsymbol{D}) = 0$$

where:

- *J* is the current density, σ**E**.
- *D* is the electric flux density, ε**E**.
- *E* is the electric field, -∇ϕ.

Complex quantities are represented using phasor notation.

AC Conduction Assumptions

The AC conduction field solver assumes the following conditions about field quantities:

- All time-varying electromagnetic quantities have the periodic waveform:

$$\boldsymbol{F}(t) = F_m \cos(\omega t + \theta)$$

  where all quantities must have the same value of ω, but can have different phase angles (θ).
- If a current is not a pure sinusoid, it is decomposed into sinusoidal harmonics, and solved separately at each frequency.
- The component of **E** due to time-varying magnetic fields caused by conduction currents can be neglected.

Admittance

Admittance can best be explained as the inverse of impedance, and is expressed by:

$$Y = G + j\omega C$$

where:

- ω is equal to $2\pi f$, where *f* is the frequency of the AC voltage source.
- *Y* is the admittance in Siemens/meter.
- *G* is the conductance in Siemens/meter.
- ωC is the susceptance in Siemens/meter.

Note	The matrix values displayed by the software are conductance in ohms/meter and capacitance in farads/meter (not conductance and susceptance) and therefore do not include jω.

Current Flow

To compute current flow, the AC conduction field solver uses the following relationship:

$$I = \int_A \boldsymbol{J} \bullet d\boldsymbol{A}$$

where:

- *I* is the current.
- *J* is the current density, given by **J** = σ**E**.
- *A* is the area over which the current flow is computed. It is found by sweeping the current flow line you've drawn in the xy-plane into the z direction, forming a surface. The current flow computed is the current per meter depth in the z direction.

A separate current flow value can be computed for each line you draw.

Axisymmetric Field Simulation

If you selected **Cylindrical About Z** as the geometry mode when setting the solution type, a special version of the selected field solver is used. It assumes that the 2D geometry being studied sweeps 360° around the z-axis of a cylindrical coordinate system, enabling you to model devices like solenoids and insulators that have an axis of rotational symmetry. All electric or magnetic fields in the problem must also be rotationally symmetric so that the solution in any RZ-plane is the same as any other.

Note	All cylindrical about Z models must be drawn in the XZ plane with X >= 0

Axisymmetric field simulation is available for the following solvers:

- Electrostatic
- Magnetostatic
- Eddy Current
- DC Conduction
- AC Conduction
- Transient with and without motion

Any field solution involving a long plank-shaped object is different from a solution involving an annular object — even though the same equation is solved for both types of geometries. Mathematically, the difference between cartesian and axisymmetric field solutions arises from the difference in how gradients, curls and divergences are expressed in the two coordinate systems.

- In a cartesian (xyz) coordinate system, these operators are defined as follows (where F is a scalar quantity and **F** is a vector quantity):

$$\nabla F(x,y,z) = \frac{\partial F}{\partial x}\hat{x} + \frac{\partial F}{\partial y}\hat{y} + \frac{\partial F}{\partial z}\hat{z}$$

$$\nabla \bullet \boldsymbol{F}(x,y,z) = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z}$$

$$\nabla \times \boldsymbol{F} = \begin{bmatrix} \hat{\boldsymbol{x}} & \hat{\boldsymbol{y}} & \hat{\boldsymbol{z}} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ F_x & F_y & F_z \end{bmatrix}$$

If **F** has a z-component only, the curl is defined as:

$$\nabla \times \boldsymbol{F} = \frac{\partial F_z}{\partial y}\hat{x} - \frac{\partial F_z}{\partial x}\hat{y}$$

- In a cylindrical (r, ϕ, z) coordinate system, these operators are defined as:

$$\nabla F(r,\phi,z) = \frac{\partial F}{\partial r}\hat{r} + \left(\frac{1}{r}\right)\frac{\partial F}{\partial \phi}\hat{\phi} + \frac{\partial F}{\partial z}\hat{z}$$

$$\nabla \bullet \boldsymbol{F}(r,\phi,z) = \left(\frac{1}{r}\right)\frac{\partial (r F_r)}{\partial r} + \left(\frac{1}{r}\right)\frac{\partial F}{\partial \phi}\phi + \frac{\partial F}{\partial z}z$$

$$\nabla \times \boldsymbol{F} = \begin{bmatrix} \hat{\boldsymbol{r}} & r\hat{\boldsymbol{\phi}} & \hat{\boldsymbol{z}} \\ \frac{\partial}{\partial r} & \frac{\partial}{\partial \phi} & \frac{\partial}{\partial z} \\ F_r & rF_\phi & F_z \end{bmatrix}$$

If **F** has a ϕ-component only, the curl is defined as:

$$\nabla \times \boldsymbol{F} = \left(\frac{1}{r}\right)\left[-\frac{\partial(rF_\phi)}{\partial z}\hat{\boldsymbol{r}} + \frac{\partial(rF_\phi)}{\partial r}\hat{\boldsymbol{z}}\right]$$

In the cylindrical case, an extra *r* shows up in places where there is no analogous *x* in the cartesian case, resulting in a different field solution. The calculations involving the differences in mathematics between axisymmetric and cartesian models are implicitly handled by the Maxwell and are transparent to you.

Transient Simulation

Transient Solver only.

The transient solver allows you to analyze the magnetic fields, energy, force/torque, power loss, core loss, speed, and flux of a model at various time steps of a solution over a specified period of time. This solver allows for non-sinusoidal current or voltage excitation, as well as rotational or translational motion.

Transient Simulation Assumptions

The transient solver assumes the following conditions about the problem:

- If motion occurs in the model, no motion occurs outside the band object.
- Rotational motion can be cylindrical or non-cylindrical.
- More than one object can be assigned identical motion within the band object.

Time-Dependent Magnetic Field Simulation

For a 2D XY problem, the vectors have only one component in the z-direction. The time-dependent magnetic equation is expressed as:

$$\nabla \times \upsilon \nabla \times A = J_s - \sigma \frac{\partial A}{\partial t} - \sigma \nabla V + \nabla \times H_c + \sigma V \times \nabla \times A$$

where:

- H_c is the coercivity of the permanent magnet.
- v is the velocity of the moving parts.
- A is the magnetic vector potential.
- V is the electric potential.

υ

- is the reluctivity.
- J_s is the source current density.

The transient solver applies a reference frame that is fixed with respect to the components in the model by setting the velocity, v, equal to zero. This is possible by considering the equation for the magnetic vector potential both in moving part and stationary part in their own reference frame in which velocity is always zero. Because the moving components have now been fixed to their own coordinate system, the partial time derivative becomes the total time derivative of A. Thus, the motion equation becomes:

$$\nabla \times \upsilon \nabla \times A = J_s - \sigma \frac{\mathbf{d}A}{\mathbf{d}t} - \sigma \nabla V + \nabla \times H_c$$

which is obtainable at each time step at every node in the finite element model. In above equation the motion is contained implicitly in the total derivative of A.

Stranded Conductors

Stranded conductors lack eddy current behavior and are considered to be filaments too thin to model in a practical finite element grid. Because of this, the transient solver assumes that their contribution to the current density is averaged over the area of problem region.

These filaments can be connected in parallel or series, and carry the same current based on:

$$\nabla \times \upsilon \nabla \times A = J_s$$

where $\mathbf{j_s}$, the uniformly distributed current density is given by:

$$j_s = d_f \frac{N_f i_f}{S_f \cdot a \cdot p}$$

and:

- i_f is the total terminal current flowing into a filament coil group (or winding).
- N_f is the total conductor number of the filaments in the winding.
- A is the magnetic vector potential.
- a is the number of parallel branches in the winding.
- d_f is the polarity (+1 or -1) to represent forward or return paths.
- S_f is the total area of the cross-section of the region occupied by the winding.
- p is the ratio of the original full model to the field domain to be solved.

Note that the filaments in the winding may reside at several different locations as long as they are connected in series or parallel.

The voltage seen from the terminal of the winding is the sum of the voltage across all the wires as well as any possible external impedance based on:

$$d_f \frac{N_f}{S_f \cdot a} \iint \frac{\mathbf{d}A}{\mathbf{d}t} \cdot d\Omega + R' \cdot i_f + L' \cdot \frac{\mathbf{d}i_f}{\mathbf{d}t} = u_s$$

and:

$$R' = r_{dc} + r_{end} + r_{ext}$$

$$L' = l_{end} + l_{ext}$$

where:

- r_{dc} is the total DC resistance.
- r_{end} is the end-turn resistance of the winding.
- r_{ext} is the external resistance connected to the winding.
- l_{end} *is the end turn inductance of the winding*.
- l_{ext} is the external inductance connected to the winding.
- *l* is the thickness of the model.

Note	Please note that the main component of inductance is calculated by Maxwell; end effects -when they can be evaluated- can however be included in the model to increase the overall accuracy of the simulation.

Solid Conductors

Solid conductors are large enough to model with finite elements, where skin effects depend not only on the frequency of the system, but on the location of nearby conductors.

Based on Ampere's Law, the total current density, $\mathbf{j_t}$, in the system is given by:

$$j_t = -\sigma\frac{dA}{dt} - \sigma\nabla V$$

which reduces to:

$$j_t = -\sigma\frac{dA}{dt} + \frac{\sigma}{l}V_b$$

or:

$$j_t = j_e + j_s$$

where:

- *Vb* is the voltage difference across the conductors end points.
- *Je* is the eddy current density.
- *Js* is the source current density.

Solid Conductors with Current Sources

For solid conductors with a current source, the total current is known, while the source component is unknown. The transient solver computes the current source based on the following circuit

equation for the n^{th} conductor:

$$\int_{\Omega_c}\int\left(-\sigma\frac{\mathbf{d}A}{\mathbf{d}t}+j_{\boldsymbol{s}}\right)d\Omega = \int_{\Omega_C}\int j_{\boldsymbol{t}}d\Omega = I_t$$

where:

- $\Omega_{\mathbf{c}}$ is the width cross-section of the *n*th conductor.
- I_t is the known total current.
- J_s is the source component to be solved for.
- J_t is the total current density.

Solid Conductors with Voltage Sources

For solid conductors with a voltage source, the total voltage is known, while the total current density is unknown. The transient solver computes the unknown quantities based on the following circuit equation which is derived from the solid conductor equations. The resulting equation yields:

$$\int_{\Omega_c}\int\left(\sigma\frac{\mathbf{d}A}{\mathbf{d}t}+j_{\boldsymbol{t}}\right)d\Omega = \int_{\Omega_C}\int\frac{\sigma}{l}V_b\,d\Omega$$

where:

- $\Omega_{\mathbf{c}}$ is the width cross-section of the *n*th conductor.
- V_b is the known voltage source between the two conductors.
- J_t is the total current density to be solved for.
- σ is the conductivity.
- l is the thickness of the model.
- A is the magnetic vector potential.

Translational Motion

The transient motion simulator generates translational motion solutions based on the following motion equation:

$$ma + \lambda v = F_{em} + F_{load}$$

where:

- m is the mass of the object, in kg.
- a is the acceleration of the object, in m/s^2.
- v is the velocity, in m/s.
- F_{em} is the computed electromagnetic force, in N.
- F_{load} is the external load force, in N.
- λ is the damping, in N·s/m.

Rotational Motion

The transient motion simulator generates rotational motion solutions based on the following motion equation:

$$J\beta + \lambda\omega = T_{em} + T_{load}$$

where:

- J is the moment of inertia, in kg·m^2.
- Tem is the computed electromagnetic torque, in N·m.
- T_{load} is the external load torque, in N·m.
- ω is the angular speed, in rad/s.
- β is the angular acceleration, in rad/s^2.
- λ is the damping factor, in N·m·s.

Core Loss for Transient Solvers

Core loss allows the solver to find the total core loss on an object, or group of objects due to the distribution of the electric or magnetic field in the design.

For frequency domain applications:

At a given frequency, the core loss for electrical steel is based on the following expression:

$$p = K_h B_{\max}^2 f + K_c (B_{\max} f)^2 + K_e (B_{\max} f)^{1.5}$$

where:

- K_h is the hysteresis coefficient.
- K_c is the classical eddy coefficient.
- K_e is the excess or anomalous eddy current coefficient due to magnetic domians.
- B_{max} is the maximum amplitude of the flux density.
- f is the frequency.

Similarly, the power ferrite core loss is based on the following expression:

$$p = C_m f^x B_{\max}^y$$

where:

- C_m is a constant value determined by experiment.
- f is the frequency.
- B_{max} is the maximum amplitude of the flux density.

For transient (time domain) applications, the above expressions need to be modified in the following ways:

In the frequency domain, the core loss for power ferrites is given by the above formula, where C_m, x, and y are measured based on sinusoidal $B(t)$ waveform. If $x=1$, $y=2$, and $C_m=K_h$, the formula for power ferrite is that for hysteresis loss. If $x=y=2$, and $C_m=K_c$, it corresponds to classic eddy loss. If $x=y=1.5$, and $C_m=K_e$, it becomes excess loss. Therefore, the formula of power loss for ferrite can be used to describe all individual components for core loss for both electric steel and power ferrite.

Based on the formula for ferrite core loss in the frequency domain, the following expression can be derived to compute core loss in the *time* domain:

$$P_f = C_t |H_{irr}|^{y-x} \left| \frac{dB}{dt} \right|^x$$

where H_{irr} is the irreversible component of the magnetic field strength $H_{irr}=H-H_{rev}$, with H_{rev} being the reversible component of H, which can be obtained from the BH curve based on B. To obtain H_{irr}, an equivalent ellipse is used with the property that the area of the ellipse equals the are of the original hysteresis curve. This explains why, in the process of obtaining the hysteresis loss, the $B(H)$ curve and K_h coefficient are used.

The equation of the equivalent ellipse is given by:

$$\left(\frac{H_{irr}}{H_m}\right)^2 + \left(\frac{B}{B_m}\right)^2 = 1$$

where

$$H_m = \frac{K_h \cdot B_m}{x}$$

and B_m is the amplitude of a minor BH loop which is obtained based on continuously updated history information of the magnetic flux density, which allows the algorithm to account for the contribution of minor loops to the core loss.

With H_{irr} and dB/dt, the loss can be easily evaluated. For the other components of the core loss (i.e., different combinations of x and y), unknown coefficients can be derived using the property that the formula for core loss in the time domain gives the same result as the formula for core loss applied for the frequency used in the experiment.

Phasor Notation

Time varying quantities that have the form:

$$\boldsymbol{F}(t) = F_m \cos(\omega t + \theta)$$

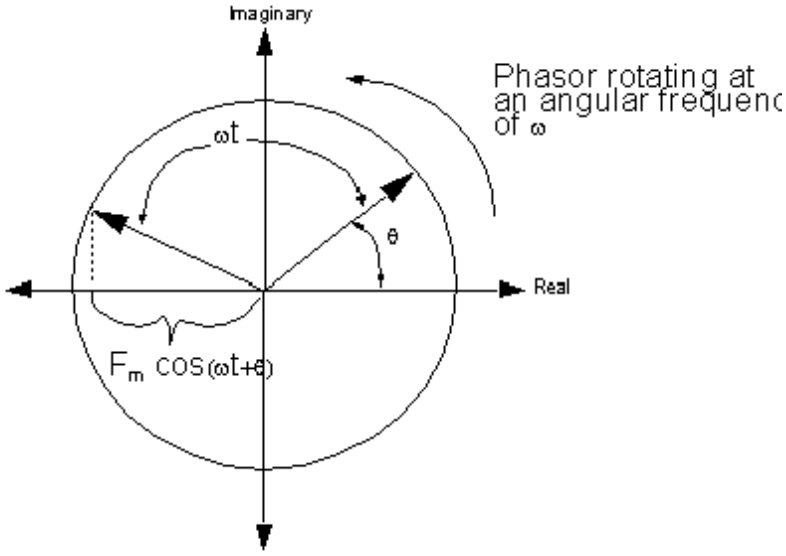
can be represented as rotating phasors in the complex plane. Using Euler's formula:

$$e^{j\alpha} = \cos\alpha + j\sin\alpha$$

If $\alpha = \omega t + \theta$, **F**(t) equals the real portion of $e^{j(\omega t+\theta)}$:

$$\boldsymbol{F}(t) = \Re\left[F_m e^{j(\omega t+\theta)}\right] = \Re[F_m(\cos(\omega t+\theta) + j\sin(\omega t+\theta))] = F_m\cos(\omega t+\theta)$$

Each time-varying quantity has the form $F_m e^{j\theta} e^{j\omega t}$. The $F_m e^{j\theta}$ component is merely a complex constant that can be represented by a stationary phasor in the complex plane. The $F_m e^{j\omega t}$ component is a complex number that depends on *t*, and can be represented as a rotating phasor in the complex plane, as shown here.



The phasor's projection on the real axis oscillates sinusoidally. It reaches a peak when parallel with the real axis, and crosses zero when parallel with the imaginary axis. Thus, a phasor with $\theta=90°$ represents a quantity that peaks 90 degrees after a phasor with $\theta=0°$.

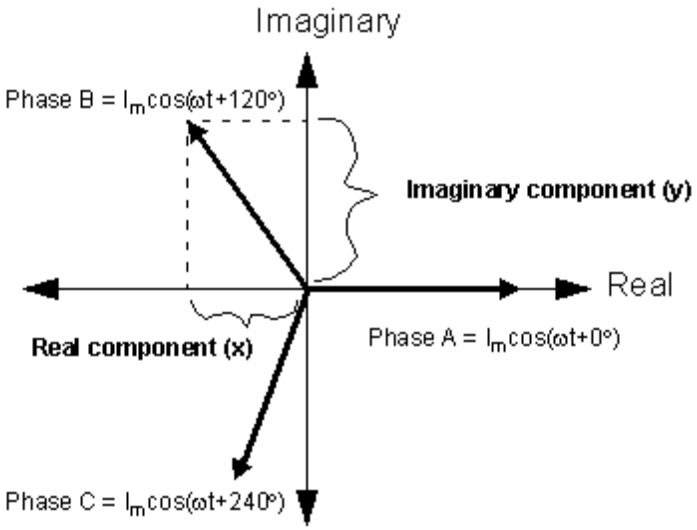
Real and Imaginary Components

Maxwell expects you to enter magnitudes and phase angles when you specify voltages, currents, and other boundary or source quantities in models where time-varying fields are to be computed (that is, eddy current, AC conduction, and eddy axial models). But when the magnitude and phase angle of a quantity are functions of position, it is easier to specify the functions in terms of real and imaginary components "x+jy" rather than in terms of magnitude and amplitude. Therefore, when you specify currents and boundary conditions as functions, the system expects you to describe the functions in terms of real and imaginary components.

The "x+jy" description of a phasor indicates that the phasor is the sum of two components — a sinusoid that peaks at $\omega t=0°$ and a sinusoid that peaks at $\omega t=90°$.

- The "x" component of the phasor is called the real component and can be represented by a phasor that lies on the real axis of the complex plane.
- The "y" component is called the imaginary component and is represented by a phasor that lies on the j-axis of the complex plane.

The real and imaginary components of a three-phase system are shown below:



The real and imaginary components are related to the magnitude and phase of a sinusoid in this way:

$$I_m = \sqrt{x^2 + y^2}$$

$$\theta = \operatorname{atan}\frac{y}{x}$$

Boundary Conditions

Boundary conditions define the behavior of the electric or magnetic field at object interfaces or edges of the problem region. They are always necessary in order to insure the uniqueness of the electromagnetic field calculation. They can also be used to:

1. Simulate structures that are magnetically isolated, electrically insulated, or electrically isolated.
2. Set the electric or magnetic potential at a surface to a constant value or a function of position, in order to define the behavior of the electric or magnetic field on that surface
3. Simulate the field patterns that would exist in a structure while modeling only part of it. To do this, you can define planes of symmetry where electric or magnetic fields are either tangential to or normal to the surface. Additionally, you can define planes of symmetry

where the field on one surface matches the magnitude and direction (or opposite direction) of the field on another surface.

4. Simulate the field patterns produced by thin resistive layers on conductors (DC conduction solver) or eddy currents with very tiny skin depths in conductors (eddy current solver), without having to explicitly draw, assign materials to, or solve for fields inside the objects in question.

Available boundary types in Maxwell 2D are listed in the following table. Select from the table to see more information about a specific boundary type or for information on how to apply it in a specific problem type.

Boundary Condition	Applying in Problem Type
Vector Potential	Magnetostatic Eddy Current Transient
Symmetry	Magnetostatic Electrostatic AC Conduction DC Conduction Eddy Current Transient
Balloon	Magnetostatic Electrostatic AC Conduction DC Conduction Eddy Current Transient
Impedance	Eddy Current
Resistance	DC Conduction
Independent/Dependent	Magnetostatic Electrostatic AC Conduction DC Conduction Eddy Current Transient

Vector Potential

The Vector Potential boundary is available for use in Magnetostatic, Eddy Current, and Transient solver type designs. Use Vector Potential boundaries to set the magnetic vector potential, $\mathbf{A}_Z$, to a

constant value on a boundary.

The potential can also be defined as a function of position using math functions. Normally, this type of boundary condition is used to specify the potential of outer boundaries. It can also be used to set the interface between two objects to a potential, modeling the presence of a very thin conductor between the objects. They are sometimes called Dirichlet boundaries.

The behavior of the magnetic field on a Vector Potential boundary depends on whether you define a constant or functional potential on the boundary. Remember that the magnetic vector potential, **A**, is defined to be a field that satisfies the equation:

$$\nabla \times \boldsymbol{A} = \boldsymbol{B}$$

Since the magnetostatic field solver assumes that **A** has a z-component only and **B** lies in the xy-plane, the relationship of **B** to **A** is given by the following:

$$\boldsymbol{B} = \frac{\partial \boldsymbol{A}_Z}{\partial y}\hat{\boldsymbol{x}} - \frac{\partial \boldsymbol{A}_Z}{\partial x}\hat{\boldsymbol{y}}$$

If $\mathbf{A}_Z$ is constant along a horizontal boundary, the partial derivatives of $\mathbf{A}_Z$ with respect to x will be zero — forcing **B** to have an x-component only, and be tangential to the boundary. Likewise, if $\mathbf{A}_Z$ is constant along a vertical boundary, the partial of $\mathbf{A}_Z$ with respect to y will be zero — forcing **B** to have a y-component only and again indicating that the field will be tangential.

In general, the magnetic field will be tangential to any boundary on which $\mathbf{A}_Z$ has been set to a constant.

If the potential is a function of position, the partial derivatives of $\mathbf{A}_Z$ with respect to x and y will not necessarily be zero. It all depends on what type of math function was used to specify the potential. Thus, **B** may not be tangential to the boundary and some flux will cross it.

In eddy current problems, the magnetic vector potential is a time-varying quantity in the form:

$$\boldsymbol{A}_Z(t) = A_m \cos(\omega t + \theta)$$

where A_m is the magnitude of the potential and θ is its phase angle — its offset from a pure cosine wave. Therefore, when specifying $\mathbf{A}_Z$ on a boundary, you must enter both its magnitude and phase. The magnitude and phase of the potential can also be defined as a function of position using math functions.

Vector Potential Boundaries in Axisymmetric Models

In axisymmetric models, **A** is assumed to have only a ϕ-component and **B** is assumed to lie only in the rz-plane. The relationship between $\mathbf{A}_\phi$ and **B** is given by:

$$\boldsymbol{B} = \frac{1}{r}\left[-\frac{\partial}{\partial z}(rA_\phi)\hat{\boldsymbol{r}} + \frac{\partial}{\partial r}(rA_\phi)\hat{\boldsymbol{z}}\right]$$

Because equipotential lines of $r\mathbf{A}_\phi$ in axisymmetric models coincide with the lines of magnetic flux, you must specify values or functions of $r\mathbf{A}$ (not **A**) when setting value boundaries.

Symmetry

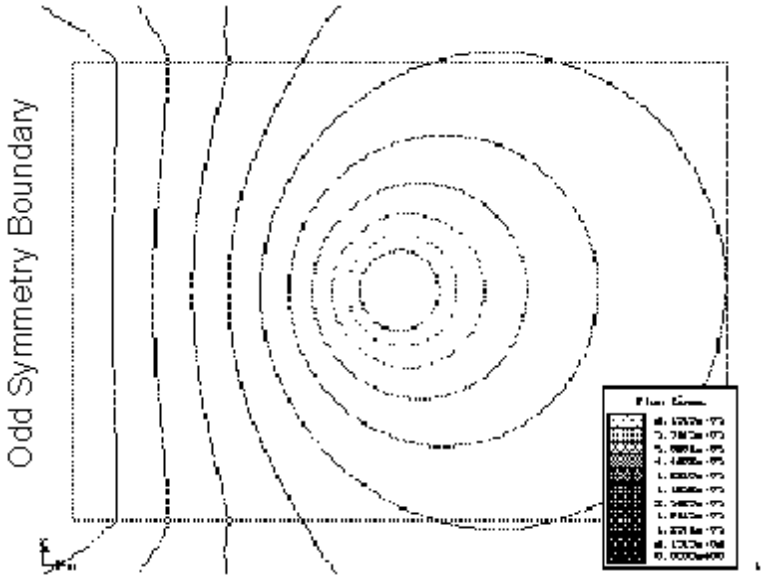
The symmetry boundary is available for use in Magnetostatic, Eddy Current, Transient, Electrostatic, AC Conduction, and DC Conduction solver type designs. A symmetry boundary models a plane of symmetry in a structure. Use this type of boundary condition to take advantage of geometric symmetry and electrical symmetry in a structure. Doing so enables you to reduce the size of your model — allowing you to conserve computing resources. Two types of symmetry boundaries — **Odd** and **Even** — are available for use.

Odd Symmetry

An odd symmetry boundary models a structure in which the signs (positive or negative) of all currents, voltages, or charges on one side of a symmetry plane are the opposite of those on the other side. In magnetic field problems, the magnetic field is tangential to this type of boundary; while in electric field problems, the field is perpendicular to the boundary and equipotential lines are tangential to the boundary.

To define an odd symmetry boundary for magnetic field problems, the simulator sets the selected edge to a vector potential boundary with a magnetic vector potential of zero — acting as a magnetic mirror to the model. For electric field problems, the simulator sets the selected edge to a voltage of zero.

For instance, the plane of symmetry shown below is modeled by an odd symmetry boundary, since the direction of the current flow in the conductor on the left side of the symmetry plane is the opposite of the current flow in the conductor on the right side of the plane (the side that is modeled):

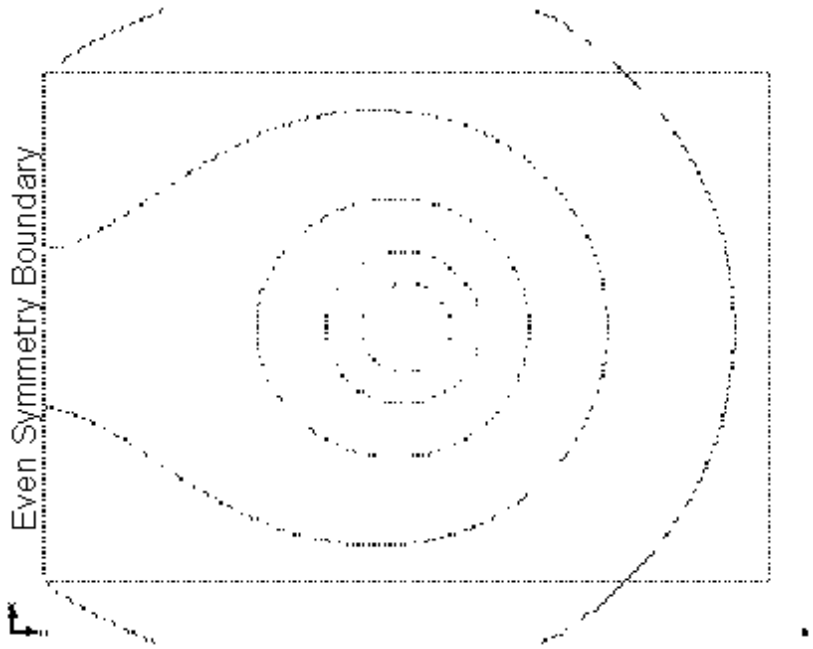


Even Symmetry

An even symmetry boundary models a structure in which the signs (positive or negative) of the currents, voltages, or charges on one side of a symmetry plane are the same as those on the other side. The magnetic field is perpendicular to this type of boundary; while the electric field is tangential to this type of boundary, and contours of equal potential are perpendicular to it.

To define an even symmetry boundary, the simulator sets the selected edge to a Neumann boundary.

For instance, the plane of symmetry shown below could be modeled by an even symmetry boundary, since the direction of the current flow in the conductor on the left side of the symmetry plane is the same as that of the current flow in the conductor on the right side of the plane (the side that is modeled):



Balloon

The balloon boundary is available for use in Magnetostatic, Eddy Current, Transient, Electrostatic, AC Conduction, and DC Conduction solver type designs. Balloon boundaries model the region outside the drawing space as being nearly “infinitely” large — effectively isolating the model from other sources of current or magnetic fields. Visualize the **background** object as extending to infinity along the edges identified as balloon boundaries.

For magnetic solution types, the magnetic vector potential, $\mathbf{A}_Z$ or $\mathbf{A}_\phi$, goes to zero at infinity. The lines of magnetic flux are neither tangential to nor normal to a balloon boundary.

For electric field solution types, two types of balloon boundaries are available:

Charge	Models the case where the charge at infinity matches the charge in the solution region, forcing the net charge to be zero. Physically, this represents an electrically insulated system. This is the default for Electrostatic solutions.
Voltage	Models the case where the voltage at infinity is zero. Physically, this represents an electrically grounded system. In most cases, the results will be very similar to those produced with the **Charge** option; however, the charge at infinity may not exactly match the charge in the drawing region.

The E-field is neither tangential to nor normal to a balloon boundary.

Impedance

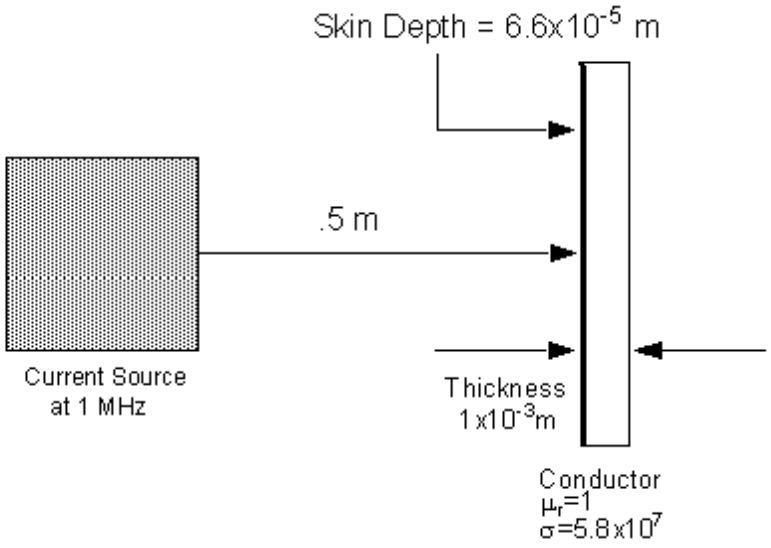
The impedance boundary is available for use in Eddy Current solutions. Impedance boundaries allow you to simulate the effect of induced currents in a conductor without explicitly computing them. The ohmic loss due to induced currents is computed from the tangential components of the H-field along the impedance boundary — the surface of the object that you are interested in.

Use this boundary condition for models where the following conditions occur:

- The skin depth in the conductor of interest is less than two orders of magnitude smaller than the dimensions of the structure. In models like this, the Maxwell 2D's meshmaker may not be able to create a fine enough mesh in the conductor to compute eddy currents.
- The magnetic field decays much more rapidly inside the conductor in the direction that is normal to the surface than it does in directions that are tangential to the surface.
- The AC current source is relatively far away from the surface where eddy currents occur, compared to the size of the skin depth.

The object itself must be excluded from the solution region by making the object a perfect conductor. When drawing the geometry, make the surface along which eddy currents are to be computed an outer surface of the problem region. Then, when defining boundaries, assign an impedance boundary to this surface. By entering the conductivity, σ, and the relative permeability, μ_r, of the object, you specify the skin depth of induced eddy currents. The simulator uses this skin depth when computing the electromagnetic field solution. It assumes that the H-field falls off exponentially inside the conductor.

For instance, suppose you want to compute eddy current losses in the conductor next to the current source shown below.



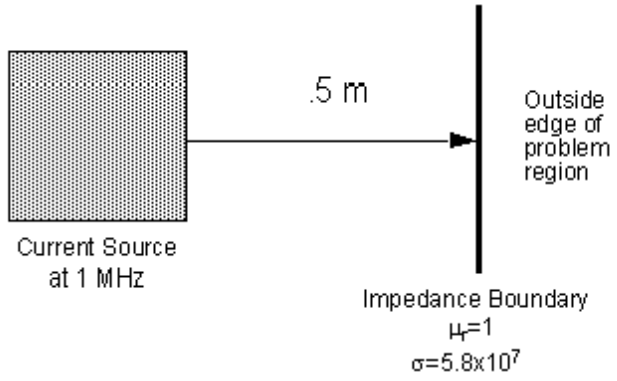
If AC current is passing through the current source at a frequency of 1 MHz, the skin depth in the conductor is given by the following relationship:

$$\delta = \sqrt{\frac{2}{\omega \sigma \mu_r \mu_0}}$$

where:

- $\omega = 2\pi f = 2\pi \times 10^6 = 6.28 \times 10^6$ radians/second
- $\sigma = 5.8 \times 10^7$ siemens/meter
- $\mu_r = 1$
- $\mu_0 = 4\pi \times 10^{-7}$ henries/meter

Substituting these values into this equation, the skin depth is found to be 6.6×10^{-5} meters. Since this is much smaller than the thickness of the conductor and the surface where currents are induced is relatively far away from the current source, an impedance boundary can be used to model the induced currents in the conductor, as shown below. The conductor itself is not included in the model; instead, the outside boundary of the model is moved to the inside surface of the conductor. This outside surface is defined as an impedance boundary, using the conductivity and permeability specified previously.



After generating a solution, you can compute the ohmic loss for the surface using the plane calculator and plot the loss density on the boundary.

For impedance boundaries, ohmic loss is given by:

$$P = \sqrt{\frac{\omega \mu_0 \mu_r}{8\sigma}} \int_{Sur} \boldsymbol{H}_t \bullet \boldsymbol{H}_t^* \, ds \quad \text{(Watts)}$$

where:

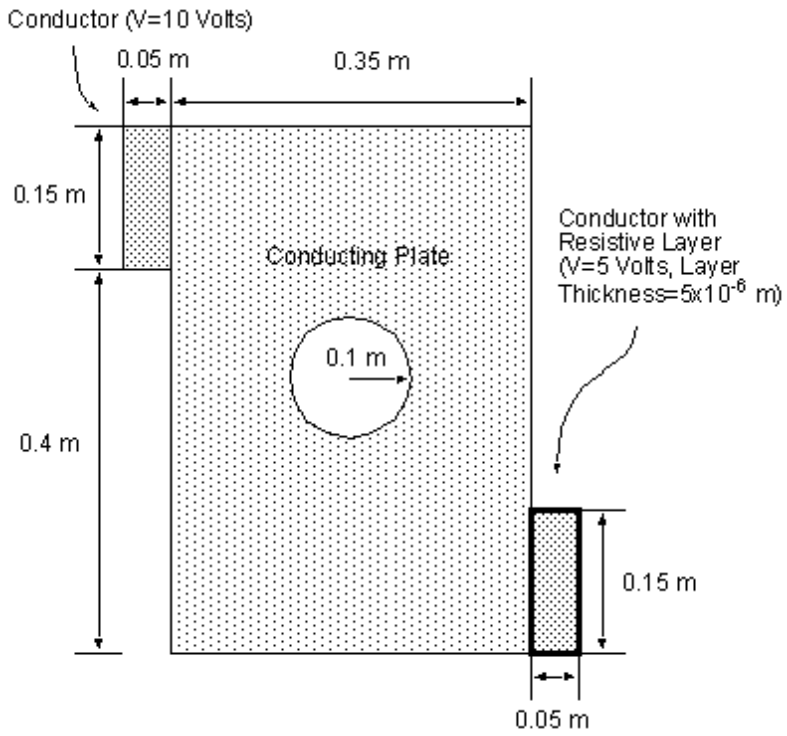
- ω is the angular frequency, which is equal to $2\pi f$.
- σ is the conductor's conductivity in siemens/meter.
- μ_r is the conductor's relative permeability.
- μ_0 is the permeability of free space, which is equal to $4\pi \times 10^{-7}$ H/m.
- $\boldsymbol{H_t}$ is the tangential component of **H** on the impedance boundary.
- $\boldsymbol{H_t^*}$ is the complex conjugate tangential component of **H** on the impedance boundary.

Note	Keep in mind that an impedance boundary approximates the effect of eddy currents acting at a shallow skin depth; it does not directly compute them. In general, the fields modeled using an impedance boundary will closely match the field patterns that would actually occur in the structure. However, the field patterns may be different at discontinuities in the surface such as corners.

Resistance

The resistance boundary is available for use in DC Conduction solutions. A resistance boundary models a very thin layer of resistive material (such as that caused by deposits or oxidation on a metallic surface) on a conductor at a known potential. Use this boundary condition when the resistive layer's thickness is much smaller than the other dimensions of the model.

For instance, in the following example, the resistive layer on the conductor is 5×10^{-6} meters thick. Since this is four orders of magnitude smaller than the dimensions of the model, use a resistance boundary on the conductor to avoid having to create a very thin object modeling the layer — which could cause problems when the Maxwell generates a mesh for the model and solves for its conduction currents.

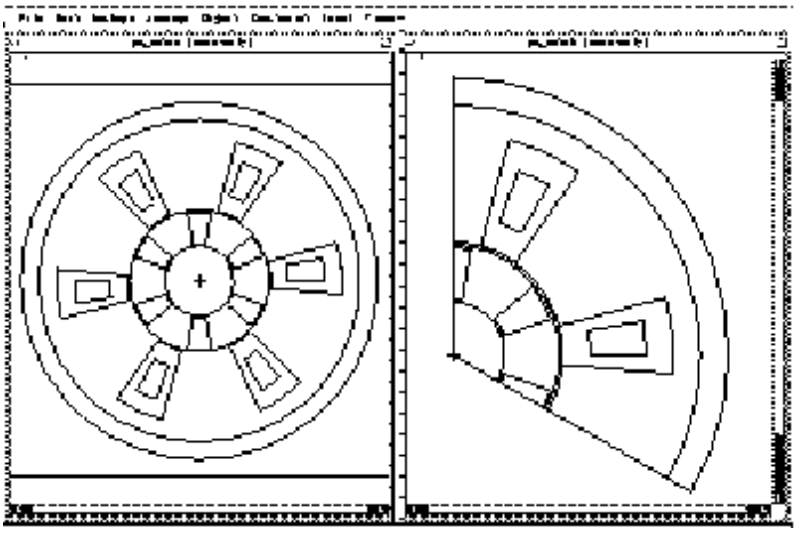


Specify the thickness and conductivity of the resistive material, and the potential of the conductor. Apply resistance boundaries only to the outside edge of the problem space. The external region touching a resistance boundary is assumed to be a PEC (perfect conductor) with a known potential value that is specified in the resistance boundary setup. For the example above, if the PEC region is included in the design model, solve inside should not be selected to exclude it from the FEM (finite element method) problem domain.

Independent/Dependent

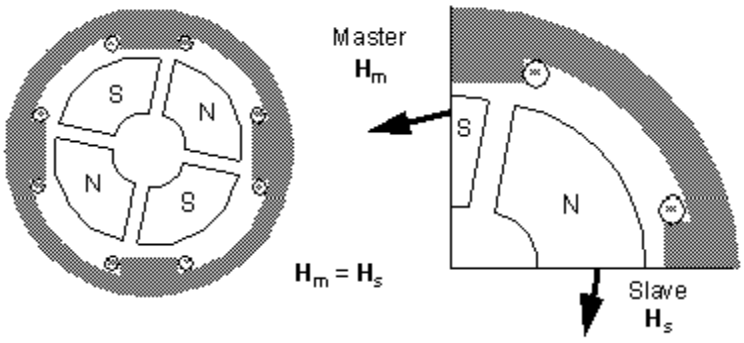
Independent/Dependent boundaries are available for use in Magnetostatic, Eddy Current, Transient, Electrostatic, AC Conduction, and DC Conduction solver type designs. Matching boundaries allow you to take advantage of periodicity in a structure. For example, the following figure shows the cross-section of a DC motor. The field in such a motor repeats itself every 120 degrees; that is, the field pattern in one third of the motor matches the magnitude and direction (or the opposite of the direction) of the field pattern in the other two thirds.

Matching boundaries force the magnetic field at each point on one boundary (the “dependent” boundary) to match the magnetic field at each corresponding point on the other surface (the “independent” boundary). Modeling one third of the structure allows you to make efficient use of the available computing resources:



To define matching boundaries, you must define both an independent matching boundary and a dependent matching boundary.

The condition that needs to be enforced, as illustrated in the following figure, is that the magnitude of the magnetic field at each point on the dependent boundary surface must match the magnetic field at each corresponding point on the independent boundary surface. The field on the dependent boundary must point in either the same direction or in the exact opposite direction as the field on the independent boundary:



One-quarter of a periodic structure (DC motor) modeled using matching boundaries.

Note that a Vector Potential (Dirichlet), Neumann or symmetry boundary cannot be used to simulate periodicity because the magnetic field is not necessarily either perpendicular or tangential to periodic surfaces. For example, in the quarter model shown above, the magnetic field is exactly perpendicular to the bounding surfaces only when the gap separating the permanent magnets is perfectly horizontal or vertical. For all other positions of the rotor, matching boundaries are required to take advantage of symmetry.

2D Sources/Excitations

To compute fields for a structure, you must define a source of charge, voltage, current, or electric or magnetic fields for your model. Assign at least one object or edge as either a source (such as a current, charge, or voltage) or a value boundary.

Permanently polarized or magnetized materials also act as sources of charge or magnetic field (respectively). If you do not identify some type of source, the Maxwell 2D will not be able to generate a solution.

The field quantities computed by each solver — and the required electromagnetic sources — are given in the following table:

Field Solver	Sources	Field Computed	Derived Field Quantities
Electrostatic	Voltages charges charge density floating	ϕ	**E**, **D**
Magnetostatic	DC current current density	$\mathbf{A}_Z$ (XY models), $\mathbf{A}_\phi$ (RZ models)	**H**, **B**
Eddy Current	AC currents current density	$\mathbf{A}_Z(\omega t)$ (XY models), $\mathbf{A}_\phi(\omega t)$ (RZ models)	$\mathbf{J}_Z(\omega t)$ (XY models), $\mathbf{J}_\phi(\omega t)$ (RZ models), **H** (ωt), **B**(ωt),
DC Conduction	DC voltages	ϕ	**E**, **D**, **J**
AC Conduction	AC voltages	$\phi(\omega t)$	**E**(ωt), **J**(ωt)
Transient	Transient voltages and currents through External Circuit connection. Current; current density; coil; end connection	$\mathbf{A}_Z$ (XY and RZ models)	**H**, **B**

where:

- **A** is the magnetic vector potential.
- **H** is the magnetic field.
- **B** is the magnetic flux density.
- ϕ is the electric potential.
- **E** is the electric field.

- **D** is the electric flux density.
- **J** is the current density.

These quantities are phasors in AC simulations.

Related Topics

Permanently Polarized Materials
External Static Magnetic Fields
Permanent Magnets
External AC Magnetic Fields

2D Voltage Sources

Voltage sources are available for the Electrostatic, DC Conduction, AC Conduction and Transient solver types. Voltage sources specify the total DC voltage (electric potential), or the magnitude and phase of the AC voltage on a conductor. Voltages can be defined as constants or as functions; however, the voltage is assumed to be uniform over the source.

Voltage sources may also be specified on a selected edge or edges.

Related Topics

AC Voltage Source
DC Voltage Source

DC Voltage Source

This type of source specifies the total DC voltage (electric potential) on a conductor. Voltages can be defined as constants or as math functions; however, the potential on a conductor is constant over the entire conductor. Note that conductors that touch should be set to the same voltage or defined as a single voltage source, since their potentials are identical.

DC voltage may also be specified on an edge or edges. Voltages can be defined as constant or as functions of position (for instance, to model a specific distribution of potential on the surface of a dielectric).

AC Voltage Source

AC voltage sources are used in AC conduction, and Transient models. All voltages in AC models are time-varying quantities in the form:

$$V(t) = V_m \cos(\omega t + \theta)$$

where V_m is the magnitude of the voltage and θ is its phase angle — the offset of the current from a pure cosine wave. Therefore, when specifying a voltage, you must enter both its magnitude and phase.

2D Current/Current Density

Current sources are available in the Magnetostatic, Eddy Current, and Transient solver types. This type of source specifies the total DC current, or the magnitude and phase of the AC current on a conductor. Currents can be defined as total current, current density, or as functions.

Current sources may also be specified on a selected edge or edges. Edge currents can be defined as total surface current or a surface current density. In the case of edge currents, Eddy current effects are not modeled, since all currents are surface currents.

Generally:

- If the total surface current is specified, the current density is assumed to be uniform.
- If the current density is specified, you may define a uniform current density or one that varies as a function of position to model specific distributions of current on the surface.

Related Topics

AC Current Source

DC Current Source

DC Current Source

This type of source specifies the DC current flowing in a conductor. You can set either the total current or the current density flowing in the object.

- If total current is specified, the current density is assumed to be uniformly distributed throughout the object.
- If current density is specified, you may define a uniform current density or one that varies as a function of position.

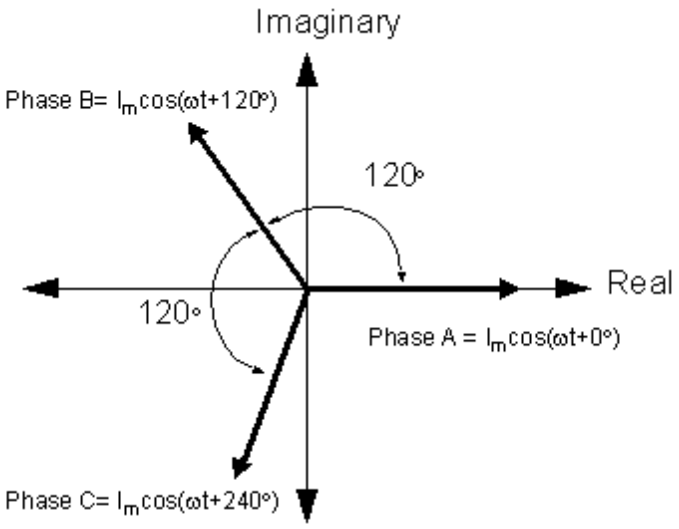
AC Current Source

Remember, all AC currents are time-varying quantities in the form:

$$I = I_m \cos(\omega t + \theta)$$

where I_m is the magnitude of the current and θ is its phase angle — the offset of the current from a pure cosine wave. Therefore, when specifying a current or current density, you must enter both its magnitude and phase.

- In a single-phase system, time t=0 is usually chosen so that the phase angle, θ, is zero — that is, the current peaks at t=0.
- In multi-phase systems involving currents that are out of phase with each other, time t=0 is usually chosen so that one current has a phase angle equal to zero. For example, phase angles in a three-phase system could be assigned as shown here:

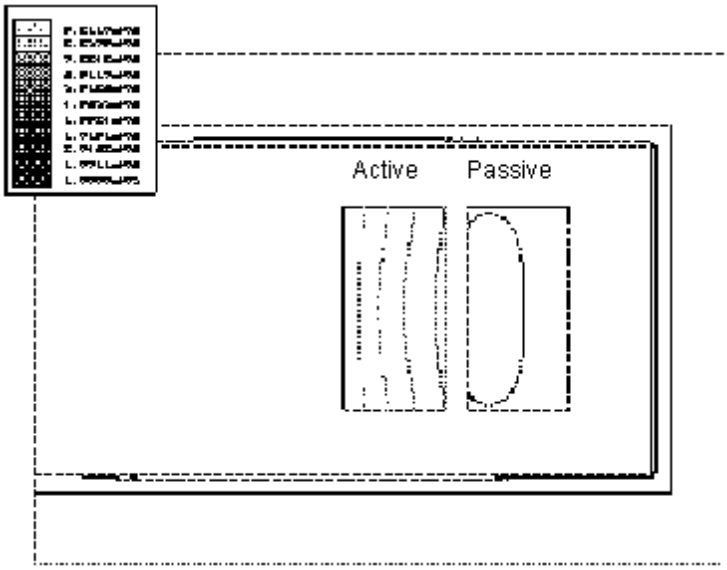


Active vs Passive Conductors

In addition, the conductors in an eddy current model can be divided into two groups:

- “Active” conductors. These conductors are connected to an external current source. Their total current is constrained to the value you specify.
- “Passive” conductors. These conductors are not connected to an external source, but current may be induced in them. Treat any conductor in which the current is constrained to zero (an open circuit) as being connected to a zero-amp current source.

Active and passive conductors are shown in the figure below. In this simple transformer model, the coil on the left is an active conductor carrying 1500 amps of current. The coil on the right is a passive conductor in which current is induced by the oscillating magnetic field. The total current is plotted.



Passive Conductors

Passive conductors can have eddy and displacement currents flowing through them, but have no component of source current. Two types of passive conductors may be defined:

- To define a passive conductor modeling a short circuit, simply assign a conducting material to the desired object. Do not assign source current to it. There are no constraints on the eddy and displacement currents flowing in this type of passive conductor. For cartesian models, visualize this type of conductor as being infinitely long and eventually looping back on itself. For axisymmetric models, visualize this type of conductor as a conducting ring that carries no source current.
- To define a passive conductor modeling an open circuit, assign a solid current source with a magnitude and phase of zero to it. Current may be induced in it, but the net current is constrained to zero amps. In cartesian models, visualize this type of conductor as an infinitely long conducting rod with no return path for current. In axisymmetric models, visualize this type of conductor as a conducting ring with a gap in it.

Current Sources for Touching Conductors

Conductors whose surfaces touch are assigned sources as follows:

- If they are not assigned the same material, these conductors must be defined as a parallel source. Otherwise, they will behave as if they are separated by a thin layer of insulating material.
- If they are assigned the same material, these conductors may be defined either as a parallel source or as grouped conductors assigned a solid source.

This distributes current appropriately across the surfaces of the conductors.

Current Sources for Perfect Conductors

This describes the case in which all current in a perfect conductor flows only on the surface of the conductor. Magnetic fields cannot penetrate this type of conductor. You can only specify the total DC current when defining a perfect conductor as a current source.

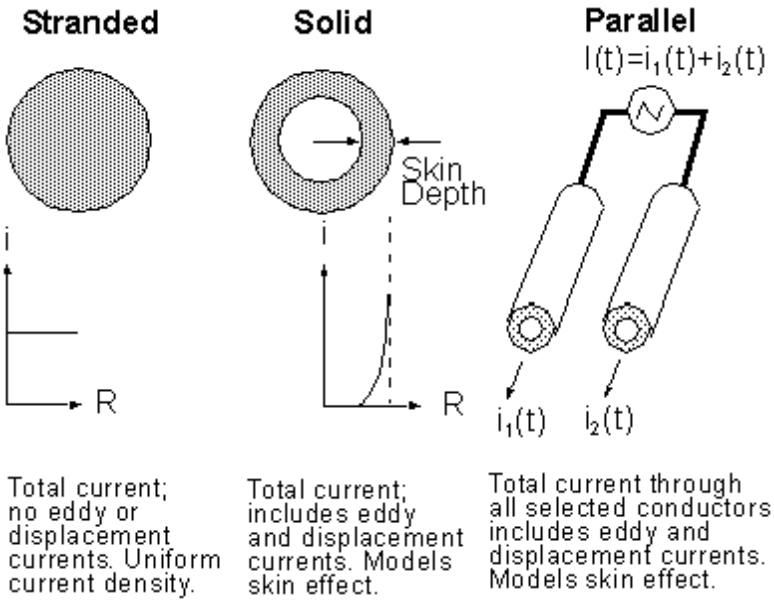
A perfect current source specifies the magnitude and phase of the AC current flowing through a perfect conductor. All currents in perfect conductors are surface currents, simulating the conductor's behavior at very high frequencies. You can only specify the magnitude and phase of the total current.

Solid, Stranded, and Parallel Current Sources

These types of sources specify the magnitude and phase of the AC current flowing through a conductor. Solid current sources model eddy and displacement currents in a solid conductor. The amount of eddy current and displacement current — as well as the amount of source current — are included in the total current you specify.

- Stranded current sources model current as being carried on strands within a conductor. They can be used to model conductors made up of many individual insulated turns, all small enough so that eddy currents can be neglected. Eddy currents and displacement currents are not computed inside the conductor. Either the total current or the current density may be specified. A uniform current density is assumed throughout the conductor, unless a functional current density is defined.
- Parallel current sources connect two or more conductors in parallel to an outside source. The total current flowing through *all* selected conductors (including eddy and displacement currents) is specified. However, the current flowing through individual conductors in the parallel group is unconstrained and its value is computed during the solution.

The differences between each type of AC current source are shown below:



For solid and parallel current sources, the current you specify is the total current in the conductor:

$$I_{Total} = I_{Source} + I_{Eddy} + I_{Displacement}$$

where:

- I_{Total} is the total current flowing through the source. It satisfies Ohm's law with the potential seen by the source.
- I_{Source} is the current due to the potential difference generated by the external source. It is the current that the source would supply if you reduced the potential difference by the back EMF produced by the eddy and displacement currents in the conductor.
- I_{Eddy} is the eddy current induced in the conductor due to time-varying magnetic fields penetrating the conductor.
- $I_{Displacement}$ is the displacement current due to time-varying electric fields in the conductor. It becomes significant only at very high frequencies.

For stranded current sources, the current you specify is the total source current (or source current density), $I_{total} = I_{source}$. Eddy current and displacement current effects are neglected.

2D Charge/Charge Density

This type of source specifies the total charge or charge density on a non-conducting object.

- If the total charge is specified, charge is assumed to be uniformly distributed throughout the interior of the object.

- If a constant value for the charge density is specified, charge is assumed to be uniformly distributed throughout the object. The charge density can also be specified as a function of position to model a distribution of charge that varies inside the object.

This type of source may also specify the charge on a selected edge or edges. It is used primarily to assign surface charges to non-conductors. The “surfaces” being referred to are those created by extending the edge in the z direction (cartesian models) or revolving it around the z-axis (axisymmetric models). Specify either the total charge or the charge density.

If the total charge is specified, charge is assumed to be evenly distributed on the selected surface.

If the charge density is specified, you can define either a uniform charge density or one that varies as a function of position — to model specific distributions of charge on the surface.

The electrostatic field simulator computes the electric potential on the edge during the solution.

2D Floating

This type of source specifies the total charge on a conductor, identifying it as a floating conductor. Charge is assumed to be evenly distributed on the object’s surface. Its value can be defined as a constant or as a function of position; however, charge is distributed over a conductor so that the electric potential is constant throughout the conductor. Because of this, the E-field is equal to zero in this region and no solution is computed inside the conductor.

External Circuit Connections

You may define transient current sources and transient voltage sources using the **Maxwell Circuit Editor**. This allows you to define an external circuit netlist which can be connected to the 2D model to act as a source in the time domain.

To define an external connection, the general procedure should be followed:

1. Create sheets objects to be used as terminals.
2. Assign the **Coil Terminal** excitation to the sheet objects.
3. Create a **Winding**, provide a unique name for the winding, and specify **External Circuit** as the winding type. Select either **Solid** or **Strand** as the type of conductor for the external connection.

Note	You cannot mix solid and stranded sources in the external circuit.

4. Choose **Winding**, and add the **Coil Terminals** to the winding using the **Add Terminals** dialog box.
5. Launch the **Maxwell Circuit Editor** from the Windows Start menu.
6. Add a **Winding** element to the circuit with the same name as given to the winding in the Maxwell design.
7. Add additional circuit element to the schematic to create the external source circuitry for the winding.

Note	The transient solver accepts only the following subset of elements available in Schematic Capture: resistors, inductors, capacitors, diodes, switch on voltages, switch on currents, current sources, voltage sources, current meters, and voltage meters.

8. **Export** the circuit to a netlist, and **Save** and **Exit** the **Maxwell Circuit Editor**.
9. In the Maxwell design, use the **Edit External Circuits** command to **Import** the netlist.

The external circuit is now ready for use in the Maxwell design.

References for Electric or Magnetic Potential

You must specify a reference for electric scalar potential or magnetic vector potential that Maxwell 2D can use when computing fields. To do so, assign one of the following boundary or source types to at least one surface in your model:

- Value boundary
- Voltage source
- Odd symmetry boundary
- Balloon boundary

If you do not set a reference for electric or magnetic potential, the model is not uniquely defined and an error message appears when you try to generate a field solution. This problem usually occurs when you set up:

- Electrostatic problems that contain only charge sources. The electrostatic field solver requires that a reference voltage be defined in order to compute the electric potential (and from it, the electric field) in the problem region.
- Magnetostatic and eddy current problems that contain only current sources. These solvers require that a reference value of $\mathbf{A}_Z$ or $r\mathbf{A}_\phi$ reference value be set in order to compute the magnetic vector potential (A_Z) and from it, the magnetic field in the problem region.

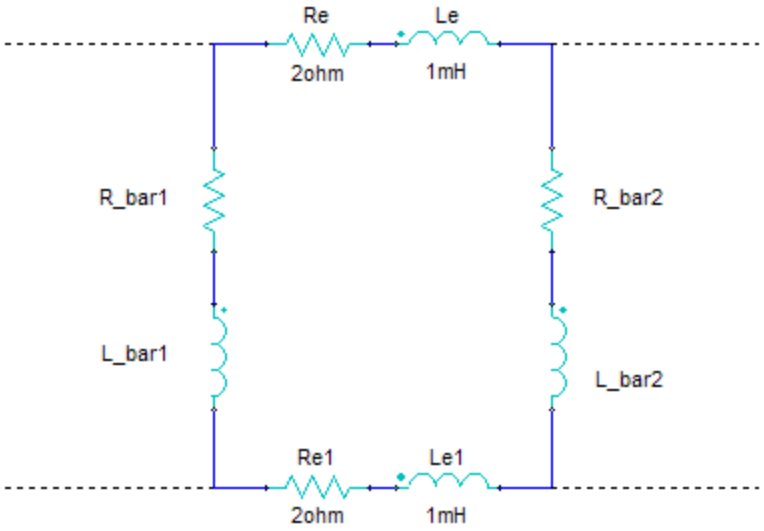
End Connections in 2D Transient and Eddy Current

Choose this command to assign an end connection to a group of solid objects. This causes all objects in the group to be connected electrically in parallel using a finite resistance and inductance between adjacent objects.

End connections are primarily used in passive conductors (with no source current assigned) when modeling cylindrical squirrel cage induction motors.

For evenly distributed conductors, the End Resistance, R_e, and the End Inductance, L_e, are assigned between the ends of each conductor pair.

Note	• Re and Le are the values on one end of the squirrel cage as shown in the schematic below. Therefore in Maxwell 2D, you would enter 2ohm and 1mH for this squirrel cage. • The current reported as EndConnection1 is the current in the first bar located counter clock-wise from the X axis. • All objects included in End Connection automatically have Eddy Effects enabled.



For unevenly distributed conductors, the user supplied resistance and inductance values are scaled such that the supplied values are used directly for the conductor pair with the smallest angle between them. All other conductor pairs are assigned values that are scaled up according to the following equations:

$$r_{eij} = r_e(\theta_{ij}/\theta_{smallest})$$
$$L_{eij} = L_e(\theta_{ij}/\theta_{smallest})$$

Where θ_{ij} is the angle between the conductor *i* and conductor *j*.

Functional Boundaries and Sources

Functional boundaries and sources have defined by math functions, and are used:

- To model distributions of charge, current or voltage that vary as a function of position.
- To model external fields that vary as functions of position.
- To define voltage, current, charge or boundary values as variables to be used in a parametric sweep.

29 - Maxwell 3D Technical Notes

Maxwell® 3D is an interactive software package that uses the finite element method (FEM) analysis to simulate and solve three-dimensional electromagnetic field problems. Maxwell integrates with other Ansys software to perform complex tasks while remaining easy to use.

Introduction to Maxwell

Maxwell 3D® supports the following five types of solutions:

- **Electric** 3D fields, which can fall in one of three categories:
  - Electrostatic 3D fields in dielectrics caused by a user-specified distribution of voltages and charges. Additional computed quantities you can specify include torque, force, and capacitances.
  - Electric 3D fields in conductors, characterized by a spatial distribution of voltage, electric field, and DC current density. The main additional quantity in this case is power loss.
  - A combination of the first two with conduction solutions being used as boundary conditions for an electrostatic problem.
- **Magnetostatic** linear and nonlinear 3D fields caused by a user-specified distribution of DC current density, voltage, permanent magnets, or externally applied magnetic fields. Additional computed quantities you can specify include torque, force, and inductances (self and mutual).
- **Harmonic (Eddy current)** (sinusoidal variation in time) steady state 3D magnetic fields with pulsation-induced eddy currents in massive (solid) conductors caused by one of the following: by a user-specified distribution of AC currents (all with the same frequency but with possibly different initial phase angles), or by externally applied magnetic fields.

  The electric field solution is correct *inside* a conductor due to Ohm's law and self-consistent current densities (including current-bunching in solid conductors). *Outside* a conductor, Maxwell 3D's electric field permits some wave effects, such as propagation to a radiation boundary, but is limited in application and will ignore capacitance effect along source conductors (no charge-bunching). In general, there is no change in the electrical phase angle of the applied current source along the length of a conductor.
- **Transient Magnetic** (time domain) 3D magnetic fields caused by permanent magnets and windings supplied by voltage and/or current sources with arbitrary variation as functions of time; electrical circuits will be connected with the windings. Rotational or translational motion effects can also be included in the simulation.
- **Transient Magnetic with A-Phi Formulation** (time domain) 3D magnetic fields caused by permanent magnets, voltages, currents, and windings supplied by voltage and/or current sources with arbitrary variation as functions of time; electrical circuits can be coupled with the field solution via the windings. Displacement current (capacitive) effect can also be included in the simulation.
- **Transient Electric** (time domain) 3D electric fields caused by time-varying voltages, charge distributions, or current excitations in inhomogeneous materials. The transient electric field simulator computes time-varying electric fields. Electric potential is the solution quantity.

The complex functionality built into the Maxwell 3D solvers is accessed through the main user interface (called the desktop). With the version 11 interface and later, you can model the problem in a fairly arbitrary order (rather than following the steps in a precise order as was required in previous versions of Maxwell).

This flexibility allows experienced users to develop a modeling style that suits their preferences. Once the model is created, the automated Maxwell solution sequence takes over and fully controls the solution process without any interaction from the user. When the solution becomes available, the user can perform a variety of post-processing tasks as required by the design application.

For users new to electromagnetic field simulation, Ansys recommends the following sequence of modeling steps:

1. Based on your application, choose the type of electromagnetic analysis to be performed.
2. Draw the geometry of the model using the drawing space provided by the Modeler menu and Draw menu commands available through the Maxwell desktop interface.
3. Assign the material properties to all solid objects in the model, and define new material properties if materials in the default library do not provide the needed material.

Note	**Caution**: Always make sure the material properties assigned to an object correspond to the real properties of the materials in the electromagnetic device that is being simulated. Material properties supplied in the default library are generic properties and may not always be substituted for actual properties.

4. Specify the field sources (excitations) and boundary conditions for your unique solution.
5. Define additional global parameters that you want to calculate (such as force, torque, inductance/capacitance, etc.).
6. Define mesh operations for special applications (such as seeding in areas/objects of interest).
7. Specify solution options.
8. Start the solution process.
9. When the solution becomes available, perform post processing, such as plotting field quantities and calculating expressions.

The basic modeling entity is the design (model). The next level up is the project. A project is a collection of one or more designs (models) that is saved in a single *.aedt file. A new project is automatically created when Ansys Electronics Desktop is launched. As many Maxwell designs as needed can be added to a single project.

When setting up a Maxwell design, specific commands are available via the menu bar or ribbons using the mouse. You can also work directly in the project tree (the Project Manager area) of the desktop.

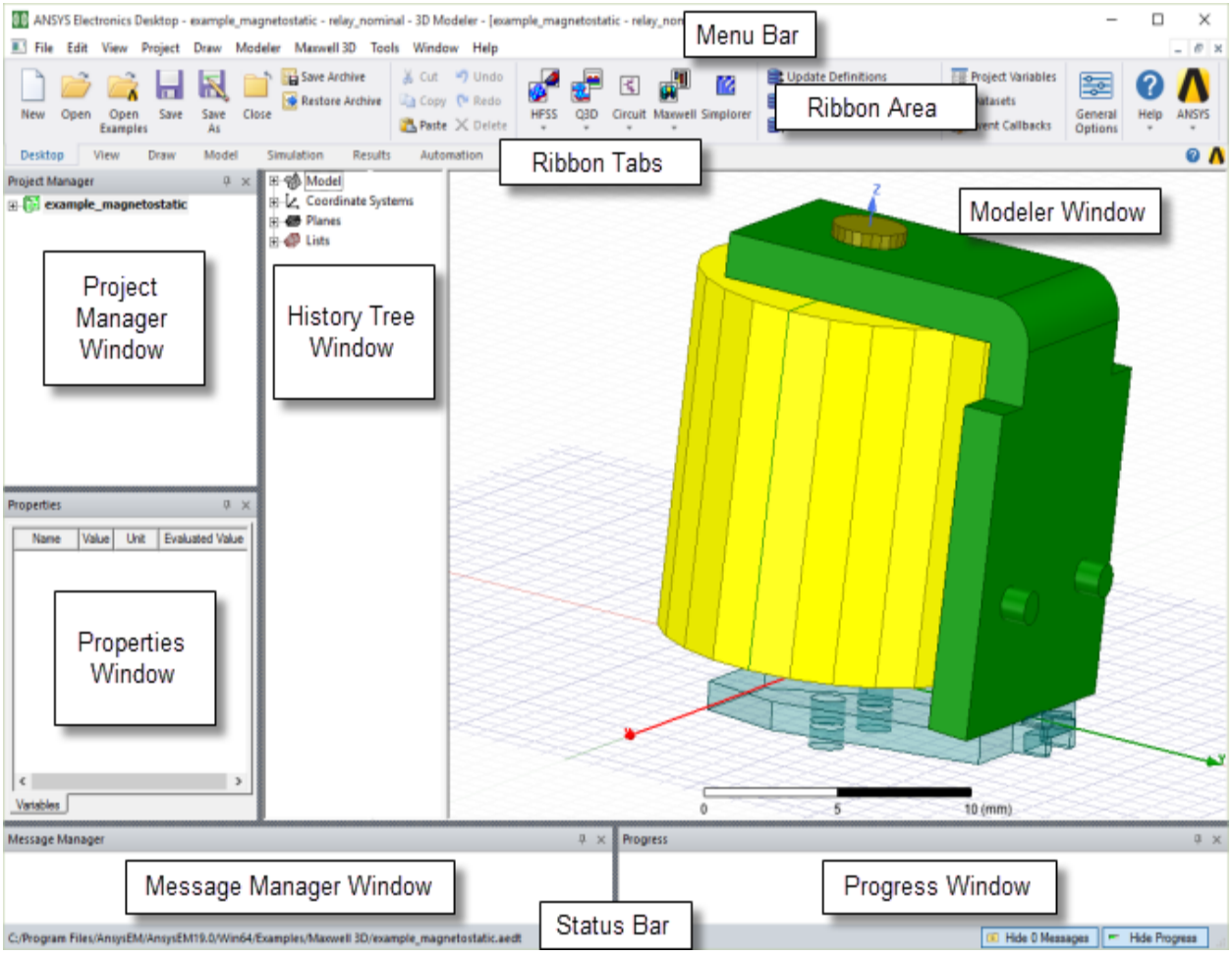
Mouse Button Usage

In general the left mouse button is used to select items (menu commands, objects, properties, etc.). Double-clicking the left mouse button on an item opens a properties window and allows you

to visualize settings and edit properties. Right-clicking is context-dependent and, in general, allows you to perform context-specific operations (such as assigning boundary conditions or excitations or performing field plots on previously selected geometry entities, exiting the zoom mode or draw mode, etc.).

Maxwell Desktop

A brief description of the desktop workspace is provided below in a graphical format. The desktop commands are located on the menu bar. The ribbons contains graphical symbols for the widely used commands. The results of the modeling operations (as far as the geometry is concerned) are reflected in the 3D modeler window and in the history tree window. The property window displays the attributes of selected objects (when the attribute tab is selected) or command characteristics (parameters) if the command tab is selected. The main modeling phases are accessible through the project manager. The message window displays messages (info, warnings, and errors) while the progress window shows the progress of the current solution process. The status bar is context dependent and typically contains info guiding the user in the modeling process (displays current status or expected input from user in the process of executing a command).

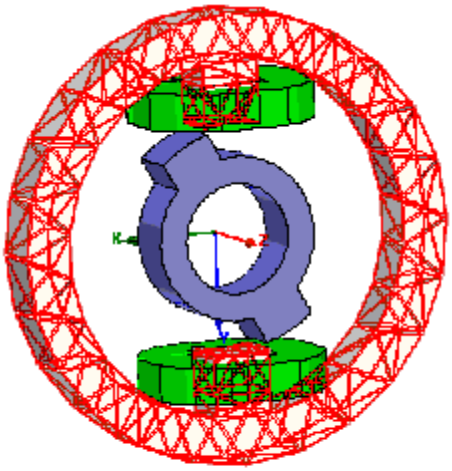


Finite Element Analysis

Finite element analysis (FEA) is a very sophisticated tool widely used by engineers, scientists, and researchers to solve engineering problems arising from various physical fields such as electromagnetic, thermal, structural, fluid flow, acoustic, and others. Currently the finite element method is clearly the dominant numerical analysis method for the simulation of physical field distributions, with success not paralleled by any other numerical technique. In essence, the finite element method finds the solution to any engineering problem that can be described by a finite set of spatial partial derivative equations with appropriate boundary and initial conditions. It is used to solve problems for an extremely wide variety of static, steady state, and transient engineering applications from diverse markets such as automotive, aerospace, nuclear, biomedical, etc.

The finite element method has a solid theoretical foundation. It is based on mathematical theorems that guarantee an asymptotic increase of the accuracy of the field calculation toward the exact solution as the size of the finite elements used in the solution process decreases. For time domain solutions the spatial discretization of the problem must be refined in a manner coordinated with the time steps of the calculation according to estimated time constants of the solution (such as magnetic diffusion time constant).

Maxwell solves the electromagnetic field problems by solving Maxwell's equations in a finite region of space with appropriate boundary conditions and — when necessary — with user-specified initial conditions in order to obtain a solution with guaranteed uniqueness. In order to obtain the set of algebraic equations to be solved, the geometry of the problem is discretized automatically into tetrahedral elements. All the model solids are meshed automatically by the mesher. The assembly of all tetrahedra is referred to as the finite element mesh of the model or simply the mesh. Inside each tetrahedron, the unknowns characteristic for the field being calculated are represented as polynomials of second order. Thus, in regions with rapid spatial field variation, the mesh density needs to be increased for good solution accuracy (see also adaptive mesh refinement).



Mesh shown on the outer armature of a rotational actuator

Types of Solutions and Degrees of Freedom (DOFs)

Solving an electromagnetic field problem is always based on solving Maxwell's equations. However the process of obtaining the solution is typically based on solving a second order consequence of Maxwell's equations with the consideration of applicable constitutive equations. At the same time -- as a rule -- a subset of complete Maxwell's equations is considered according to characteristic aspects of the application. Thus for reasons of efficiency of the solution, applications are classified as electrostatic, magnetostatic, frequency domain or time domain and as a consequence a specific type of solver is used in each case. This allows the users to obtain the solution with the desired accuracy but always within the limits of the fundamental assumptions made when the application was classified along the lines of the above mentioned criteria. The guide lines that can be used to correctly identify the type of solution to use are mentioned in the following paragraphs. The unknowns for each type of solution can be different, depending on the formulation.

Size of Mesh Vs. Accuracy

There is a trade-off among the size of the mesh, the desired level of accuracy, and the amount of available computing resources.

The accuracy of the solution depends on the size of each of the individual elements (tetrahedra). Generally speaking, solutions based on meshes using thousands of elements are more accurate than solutions based on coarse meshes using relatively few elements. To generate a precise description of a field quantity, each element must occupy a region that is small enough for the field to be adequately interpolated from the nodal values.

However, generating a field solution involves inverting a matrix with approximately as many elements as there are tetrahedra nodes. For meshes with a large number of elements, such an inversion requires a significant amount of computing power and memory. Therefore, it is desirable to use a mesh fine enough to obtain an accurate field solution but not so fine that it overwhelms the available computer memory and processing power.

To produce the optimal mesh, Maxwell uses an iterative process, called an adaptive analysis, in which the mesh is automatically refined in critical regions. First, it generates a solution based on a coarse initial mesh. Then, it refines the mesh in areas of high error density and generates a new solution. When selected parameters converge to within a desired limit, Maxwell breaks out of the loop.

The Mesh Generation Process

Following is the general mesh generation process:

1. Maxwell generates an initial mesh, which includes surface approximation settings. If necessary, the mesher will automatically perform any repairs needed to recover an accurate mesh representation of a model. The solution profile will indicate when mesh repairs have been made, and the results of these repairs will be displayed per object in the mesh statistics panel.
2. Any mesh operations that were defined are used to refine the mesh.
3. Using the resulting mesh, Maxwell computes the electromagnetic fields that exist inside the structure based on the assigned excitations.
4. If you are performing an adaptive analysis, Maxwell uses the current finite element solution to estimate the regions of the problem domain where the exact solution has strong error. Tetrahedra in these regions are refined.
5. Maxwell generates another solution using the refined mesh.
6. Maxwell recomputes the error, and the iterative process (solve — error analysis — adaptive refinement) repeats until the convergence criteria are satisfied or the maximum number of adaptive passes is completed.
7. If a frequency sweep is being performed, then Maxwell solves the problem at the other frequency points without further refining the mesh. An adaptive solution is performed only at the specified solution frequency.

Note	Maxwell does not generate an initial mesh each time it starts the solution process. The initial mesh is generated only if a current mesh is unavailable.

Related Topics

Seeding the Mesh

Guidelines for Seeding the Mesh

Length-Based Mesh Refinement

Skin Depth-Based Mesh Refinement

Surface Approximation Settings

Guidelines for Modifying Surface Approximation Settings

Meshing Region Vs. Problem Region

Model Resolution

Seeding the Mesh

In Maxwell, mesh operations are optional mesh refinement settings that enable you to provide Maxwell with engineering guidance based on your knowledge of the parts of the model geometry that are critical to the structure's electromagnetic performance. Providing such guidance to Maxwell prior to beginning the adaptive analysis process can reduce (sometimes extensively) the number of passes necessary to converge upon a field solution as well as the final number of tetrahedra in the mesh for that solution. Although adaptive analysis convergence targets areas where field behavior is found, refining the mesh using more than the standard criteria, such as material characteristics, can result in finding areas of critical field behavior as soon as the first few passes are solved.

The technique of guiding Maxwell's mesh construction is referred to as "seeding" the mesh. Seeding is performed using the **Mesh** commands on the Maxwell menu.

You can instruct Maxwell to refine the length of tetrahedral elements on a surface or within a volume until they are below a certain value (length-based mesh refinement) or you can instruct Maxwell to refine the surface triangle length of all tetrahedral elements on a surface or volume to within a specified value (skin depth-based mesh refinement.) These types of mesh operations can be defined at any time. If you apply them before the adaptive solution process, they are used to refine the initial mesh after it has been generated. You can also choose to apply mesh operations without generating a solution, in which case the mesh operations are applied to the current mesh.

In a few circumstances, you may also want to define a mesh operation that modifies Maxwell's surface approximation settings for one or more faces. Surface approximation settings are only applied to the initial mesh.

Related Topics

Defining Mesh Operations

Technical Notes: The Mesh Generation Process

Guidelines for Seeding the Mesh

While seeding the mesh is not required, it is useful in the following conditions:

- Seeding the mesh inside a volume in the model geometry where regions of strong electric or magnetic fields (with strong capacitive or inductive loading) are expected. Examples include a capacitively loaded gap in a resonant structure, sharp waveguide angles or corners, or gaps between multi-coupled lines in filter structures.
- Seeding the mesh on every face of higher aspect ratio boundaries, such as long PCB traces or on the surfaces of long wires. Spacing the mesh points roughly equal to the

trace width of the wire diameter enables you to more accurately capture the behavior of the high-aspect structure from the first adaptive pass.

Related Topics

Defining Mesh Operations

Length-Based Mesh Refinement

When you request length-based mesh refinement, you instruct Maxwell to refine the length of tetrahedral elements until they are below a specified value. The length of a tetrahedron is defined as the length of its longest edge.

You can specify the maximum length of tetrahedra on faces or inside of objects. You can also specify the maximum number of elements that are added during the refinement. When the initial mesh has been generated, the refinement criteria you specified will be used to refine the initial mesh.

Related Topics

Assigning Length-Based Mesh Refinement on Object Faces

Assigning Length-Based Mesh Refinement Inside Objects

Skin Depth-Based Mesh Refinement

When you request skin depth-based mesh refinement, you instruct Maxwell to refine the surface triangle length of all tetrahedral elements on a face to within a specified value. A layered mesh is created based on the surface mesh. The layers are graded based on the skin depth and number of layers you specify.

During skin depth-based mesh refinement, Maxwell creates a series of layers that are planes parallel to the object face, and that are spaced within the specified skin depth. For each point on the surface of the face, a series of points (P0, P1, P2, ..., Pn) are added to the mesh, where n is the number of layers. P0 is the point on the surface and the distance from P0 to Pn is the skin depth. The points are spaced in a non-uniform manner, with the distance between them decreasing in a geometric progression, as you move from Pn to P0.

For example, if

Skin Depth:	12 mm
Number of Layers of Elements:	4

then

Distance [P0,P1]:	0.8 mm.
Distance [P1,P2]:	1.6 mm.
Distance [P2,P3]:	3.2 mm.
Distance [P3,P4]:	6.4 mm.
Distance [P0,P4]:	0.8 + 1.6 + 3.2 + 6.4 = 12 mm

The skin depth-based refinement first satisfies the surface triangle edge length criterion, then introduces the series of points to each additional layer. If a limit has been placed on mesh growth, one of the following happens:

- The limit is set high enough to complete the skin depth refinement.
- The limit is set high enough to satisfy the surface triangle edge length criterion, but not high enough to complete the depth seeding.
- The limit is not set high enough to satisfy even the surface triangle edge length criterion.

Because refining by skin depth can add many seeding points, you should first refine the surface of the object using length-based mesh refinement to obtain an accurate count of the number of points Maxwell will add when refining by skin depth. This allows you to reach the surface edge length criterion and approximate the number of elements in the mesh and the number of points on the surfaces before proceeding to skin depth seeding.

The refinement criteria you specified are used to refine the current mesh.

Related Topics

Assigning Skin Depth-Based Mesh Refinement on Object Faces

Surface Approximation Settings

Object surfaces in Maxwell may be planar, cylindrical or conical, toroidal, spherical, or splines. The original model surfaces are called *true surfaces*. To create a finite element mesh, Maxwell first divides all true surfaces into triangles. These triangulated surfaces are called faceted surfaces because a series of straight line segments represents each curved or planar surface.

For planar surfaces, the triangles lie exactly on the model faces; there is no difference in the location or the normal of the true surface and the meshed surface. When an object's surface is non-planar, the faceted triangle faces lie a small distance from the object's true surface. This distance is called the *surface deviation*, and it is measured in the model's units. The surface deviation is greater near the triangle centers and less near the triangle vertices.

The normal of a curved surface is different depending on its location, but it is constant for each triangle. (In this context, "normal" is defined as a line perpendicular to the surface.) The angular difference between the normal of the curved surface and the corresponding mesh surface is called the *normal deviation* and is measured in degrees.

The *aspect ratio* of triangles used in planar surfaces is based on the ratio of circumscribed radius to the in-radius of the triangle. It is unity for an equilateral triangle and approaches infinity as the triangle becomes thinner.

You can modify the surface deviation, the maximum permitted normal deviation, and the maximum aspect ratio of triangles settings on one or more faces at a time in the **Surface Approximation** dialog box. (Click **Maxwell3D>Mesh>Assign Mesh Operation>Surface Approximation**.)

The surface approximation settings are applied to the initial mesh.

Note	For the initial mesh, all the vertices of the triangles lie on the true surfaces. During mesh refinement, all the points are first added to the faceted surface for all products, not to the true surfaces. Then, for all Maxwell projects, and for curvilinear meshing enabled HFSS and HFIE projects, the points on the faceted surfaces will be moved to the curved surface locations when such movement does not degrade the quality of the mesh.

Related Topics

Modifying Surface Approximation Settings

Technical Notes: Guidelines for Modifying Surface Approximation Settings

Technical Notes: The Mesh Generation Process

Guidelines for Modifying Surface Approximation Settings

If you intend to modify the surface approximation settings for an object face or faces, keep the following guidelines in mind:

- When necessary, override the default surface approximation settings to represent curved surfaces more accurately. More accurate representation will increase the mesh size and consume more CPU time and memory. The default settings are adequate for most circumstances.
- If you want to obtain a faster solution by using a cruder representation of curved surfaces, set the coarser setting for the whole object, not just a single face.
- It is difficult for Maxwell to satisfy aspect ratio demands if the aspect ratio value is set close to 1 because an arbitrary shape cannot be filled with only equilateral triangles. Therefore, setting the aspect ratio to 1 can lead to unreasonably large meshes. Maxwell limits the aspect ratio to 4 for planar objects and 1.2 for curved objects.

Related Topics

Modifying Surface Approximation Settings

Technical Notes: Surface Approximation Settings

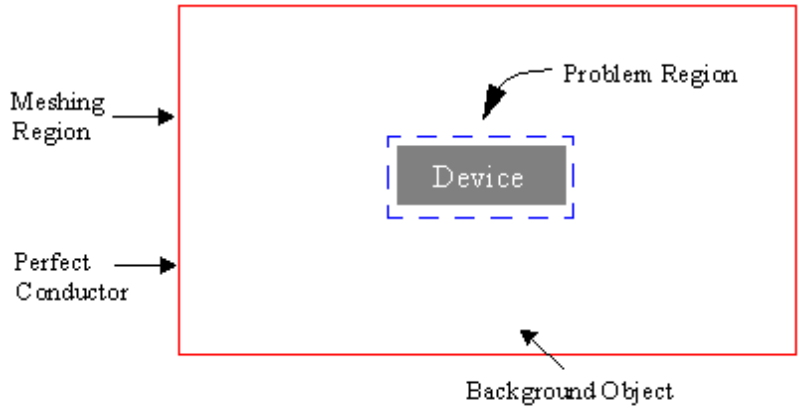
Meshing Region Vs. Problem Region

Maxwell distinguishes between the problem region and the meshing region. The *problem region* is the region in which the solution is generated and the mesh is refined. The *meshing region*, which includes the problem region, is the area in which an initial mesh is generated. After an initial mesh is generated, the mesh is refined only in the problem region.

The problem region encompasses an area that is just large enough to include the entire design, but no larger. Maxwell automatically defines the problem region during the solution process. If you are interested in effects outside of the structure, such as radiated effects, then you can create a virtual object to expand the size of the problem region to include these areas.

The meshing region, like the problem region, is a box that completely encloses the structure. However the meshing region must be at least 10 times larger than the model. The part of the meshing region not occupied by objects is considered to be the *background* object. The background extends to the boundaries of the meshing region and fills in any voids not occupied by objects. Since the background object is defined as a perfect conductor, no solution is generated inside the background even though an initial mesh is generated for it. Maxwell automatically defines the meshing region during the solution process.

The problem region and the meshing region are illustrated below.



Model Resolution

Model Resolution is a setting that determines the smallest details of a model that the mesher should capture and represent in the mesh.

Many times the analysis starts with the geometry already drawn in a different tool for different purpose. Some tools are designed for manufacturing and the resulting models contain lots of extra details not needed for electromagnetic analysis. If the user removes such details in the original tool the results will be better. But if the user does not have access to the original drawing tool or redrawing the model without these details is not possible, **Model Resolution** is another way to remove the details from analysis.

When the user sets the model resolution length to be L, the mesher will start with a surface representation of the model accurate to the modeler's tolerance limit. Then it will progressively remove edges, move points, merge points etc., within the allowable model resolution limit and simplify the surface mesh. During this process, tiny fillets, rounds, and chamfer protrusions are removed.

Other common model translation anomalies are also handled using **Model Resolution**. For example, some geometry engines will blindly export all of the surfaces as splines. When a user imports such a model for analysis, it would result in very large number of triangles. If the surface can be represented by a smaller set of triangles using Model Resolution, this procedure would reduce the number of triangles in the surface mesh.

The user can start with a model resolution length around 0.1*wavelength. If the model resolution length chosen by the user is too large, the mesher will detect it and report it as an error. The model resolution length is specified in the user units of the modeler. It can be set on selected bodies only. The default value is 100* the tolerance limit of the modeler.

Related Topics

Modifying Model Resolution

Electric Field Calculation

The Electric field solver uses the electric scalar potential values as DOFs. There are two basic types of static electric field solutions, as well as the combination of these two:

- For the electrostatic solution, it is assumed that all objects are stationary (velocity is zero everywhere in the space where the electrostatic field is calculated), there is no time

  $$\frac{\partial}{\partial t} = 0$$

  variation of any of the electromagnetic quantities ( ) and there is no current flow in conductors (conductors are in electrostatic equilibrium) and thus Joule losses are zero everywhere. All conductors are considered to be perfect and equipotential such that there is no electric field inside conductors. The unknowns (DOFs) are electric scalar potentials at all the nodes of the tetrahedra of the mesh. The distribution of conductivity in the space of the problem is irrelevant, only the distribution of permittivity is relevant for the solution process.

Note	For the electrostatic solution, if two or more conductors touch each other, they should be united into one object by using the Modeler Boolean operation "Unite" before analysis.

- For the DC current flow solution, only the conductors (with electric conductivity σ > Insulator/Conductor Threshold) are considered in the solution process. The DOFs are also the electric scalar potentials at the nodes of the mesh. Here the distribution of electric conductivity is relevant for the solution process while the permittivity is irrelevant. There is DC current flow in conductors (conductors are not equipotential) and there is non-zero Joule loss (ohmic power loss) in the conductors (which are part of an excited current path).

The conductors are considered to be immobile (velocity is zero everywhere). The magnetic field associated with the DC current flow is not calculated and is totally de-coupled from the calculated electric field distribution in conductors.

- The combination of the DC current flow and electrostatic solution is based on the division of the arrangement into conductors and insulators. Conductors are those objects whose electric conductivity σ is greater then or equal to the Insulator/Conductor threshold, and insulators are those whose conductivity σ is less then the Insulator/Conductor threshold. The solution of such problems is performed in two steps: first the DC conduction problem in the conductors is computed, then the electrostatic solution in insulators is calculated using the electric scalar potential of the conductors as a voltage boundary condition.

For the electric field solution, there are ten nodal DOFs (electric scalar potential unknowns) associated with each tetrahedron at each of the four vertices and at each of the six mid-edge nodes. Therefore, a quadratic approximation of the electric scalar potential inside each tetrahedron can be obtained.

The **electrostatic field solver** solves a Poisson type partial differential equation for the electric potential unknown, Φ, with appropriate boundary conditions:

$$\nabla \cdot (\varepsilon_r \cdot \varepsilon_0 \nabla \Phi) = -\rho_v$$

where:

$\Phi(x, y, z)$

- is the electric potential scalar function of position.

$\varepsilon_r(x, y, z)$

- is the relative permittivity, which is a function of position and can be isotropic or orthotropic.

$\varepsilon_0 = 8.854 \times 10^{-12}$

- F/m is the permittivity of a vacuum.

$\rho_v(x, y, z)$

- is the volume density of the electric charge.

Once the electric potential is obtained by solving the above second order differential equation in the domain of the problem, Maxwell's equations and the applicable constitutive equation are then used to obtain the electric field strength, $\vec{E}$, and electric flux density, $\vec{D}$, vectors, as follows:

$$\vec{E} = -\nabla\Phi$$

$$\vec{D} = \varepsilon_r \cdot \varepsilon_0 \cdot \vec{E}$$

Typical sources for electrostatic problems are net **charges** (assumed to have a uniform distribution) applied to perfect insulator model objects or on surfaces that cannot touch conductors and **voltages** (electric potential applied to perfect conductor model objects or on surfaces, also called a Dirichlet boundary condition). Additionally, a **floating** boundary condition can be applied to perfect conductors (surrounded by insulators) or to surfaces surrounded by perfect insulators.

Volume charge density is another allowed excitation, which can have a spatial distribution specified by the user and can be applied to insulating objects only that do not touch conductors.

At the interface between two dielectrics, the normal component of the electric flux density vector has no jump if the charge density at the respective interface is zero:

$$\varepsilon_1 \cdot E_{n1} = \varepsilon_2 \cdot E_{n2}$$

where the permittivity and normal electric field values are considered in the two dielectrics and are designated by the subscripts 1 and 2. Alternatively, when crossing a surface with non-zero charge density, the normal component of the electric flux density has a jump equal to the respective local superficial charge density. It follows that in dielectrics (such as air, for example) in the immediate vicinity of conductive objects (perfect conductors), the (normal) component of the electric flux density is equal to the neighboring local charge density on the metallic surface (since in metallic objects the field value is zero).

Note	When voltages are applied to conductors, the charge transferred from the DC source (s) to the conductors almost instantly receives a superficial distribution. For real (good) conductors, the time constant for the diffusion of the charge is of the order of 10^{-19}s. Therefore, for all practical instances, we consider the charge relaxation process to occur instantly. Once the charge on conductors reaches the steady state (almost instantaneously), the electric field in the neighboring dielectrics has a distribution that is

	governed by the laws of electrostatics, and the field in conductors is null while the field distribution in dielectrics is dictated - as far as material properties are concerned - by the respective permittivities and is independent of the conductivity of the electric conductors. This is why, in electrostatics, we can consider the real conductors to be perfect conductors.

Electrostatic regime of the electromagnetic field is rarely achieved in practice rigorously speaking. However very good approximations are many times possible to use for practical situations within the limits specified above.

The DC current flow solver solves the following equation:

$$\nabla \cdot (\sigma \nabla \Phi) = 0$$

where:

$\Phi(x,y,z)$

- is the electric potential scalar function of position.

  $\sigma(x,y,z)$

- is the electric conductivity (in SI it is measured in S/m), which is a function of position and can be isotropic or orthotropic.

Once the electric potential is obtained by solving the above second order differential equation in the domain of the problem, Maxwell's equations and the applicable constitutive equation (Ohm's law in a local formulation) are then used to obtain the electric field strength and the electric current density $\vec{J}$ vectors:

$$\vec{E} = -\nabla \Phi$$

$$\vec{J} = \sigma \cdot \vec{E}$$

Typical sources for DC current flow problems are **currents** applied on surfaces of conductors and **voltages** (electric potential applied to surfaces of conductors). The direction of the applied current is either "in" or "out", always normal to the respective surfaces.

Note	Multiple conduction paths are allowed. Each conduction path that has a current excitation must also have either a voltage applied or a **sink** to ensure a unique solution.

The sink is a particular excitation (without a numerical value assigned to it) that is used to ensure the divergence-free character of the calculated DC current density vector in each conduction path.

At the interface between two different conductors, the normal component of the current density vector is continuous:

$$\sigma_1 \cdot E_{n1} = \sigma_2 \cdot E_{n2}$$

For DC current flow problems, the solution is dictated everywhere inside the conductors by the distribution of electric conductivity, as far as material properties are concerned.

Charge, voltage, floating, current, and sink type excitations are called "terminal" type excitations since they can be referenced in a matrix extraction setup.

In perfect conductors, no computation occurs, and the electric scalar potential (the unknown in the problem) has a uniform value. However, in the post-processing phase, the (constant) distribution of the electric scalar potential (voltage) is correctly rendered. No solution is available in solids explicitly excluded from the solution.

The solution of any field problem is only possible if appropriate boundary conditions have been set. The boundary conditions have three purposes:

- Boundary conditions are always necessary from a mathematical perspective in order to ensure the uniqueness of the solution calculated by Maxwell.
- Boundary conditions occasionally represent a convenient way of modeling different ideal situations. For example, in order to model the field in a dielectric sandwiched between two very thin conducting objects (the thickness is at least two orders of magnitude less than the other relevant dimensions of the respective conducting objects), only the two respective surfaces need to be modeled on either side of the dielectric object. In the setup, those top and bottom surfaces carry appropriate boundary conditions, and the field in the dielectric is correctly simulated without having to draw the respective conductors.
- Boundary conditions also occasionally provide a convenient way of taking advantage of symmetry situations. In such cases, the models must have the same geometric and material properties, as well as electric symmetry.

Typical boundary conditions include voltage, floating, and insulation boundary conditions. In the case of an **insulating boundary condition**, a conductor must exist at least on one of the sides of the respective surface.

The default boundary condition applied to all faces of the boundary of the electric field (static or stationary) problem is a Neumann homogeneous condition. This condition enforces an electric field tangent behavior of the solution in the immediate vicinity of the boundaries. There is no need to apply any particular boundary condition at the interface between objects with different material

properties, as the solution automatically ensures -in average - the continuity of the tangential component of the electric field strength ($\vec{E}$) and the continuity of the normal component of the electric flux density vector ($\vec{D}$) across surfaces with zero superficial electric charge density (for electrostatic problems).

Symmetry for an Electric Field Solution

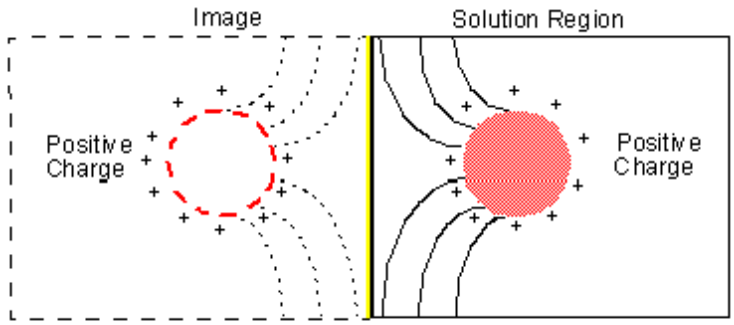
A symmetry boundary models a plane of symmetry in a structure. Use this type of boundary condition to take advantage of both geometric symmetry and electric symmetry. Doing so enables you to reduce the size of your model - allowing you to conserve computing resources. Two types of symmetry boundaries are available:

- Even Symmetry (Flux Tangential)
- Odd Symmetry (Flux Normal)

These boundaries can only be assigned to the outside edges of the solution region.

Even Symmetry (Flux Tangential) for an Electric Field Solution

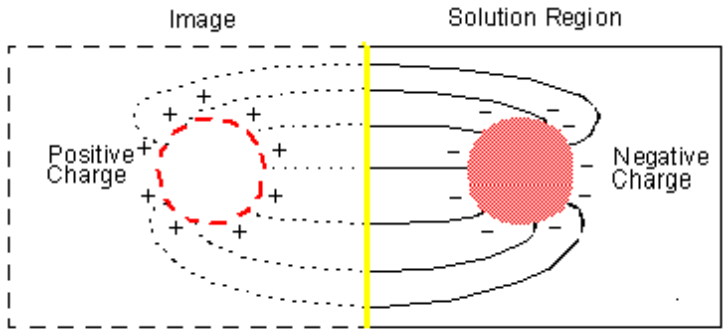
Use an even symmetry boundary to define a plane of symmetry where the signs (positive or negative) of the voltages and charges on one side of the plane are the same as those on the other side. Electric flux is tangential to the boundary and thus does not cross it. To define an even symmetry boundary, the simulator sets the selected edge to a Neumann boundary - acting as an electrical mirror to the model.



The E-field is tangential to an even symmetry boundary.

Odd Symmetry (Flux Normal) for an Electric Field Solution

Use an odd symmetry boundary to define a plane of symmetry where the signs (positive or negative) of all charges and voltages on one side of the plane are the opposite of those on the other side. Electric flux is normal to the boundary. To define an odd symmetry boundary, the simulator sets the selected edge to a voltage boundary with a potential of zero volts.



The E-field is perpendicular to an odd symmetry boundary.

Matching Boundary for an Electric Field Solution

Matching boundaries allow you to model planes of periodicity where the E-field on one surface matches the E-field on another. They are very useful for modeling devices such as motors, in which the electric field repeats every 180^o, 120^o, 90^o, or less. They enable you to model the smallest possible periodic segment of the device - reducing the amount of computing resources needed during the solution.

To set up matching boundaries, you must create an independent boundary and a dependent boundary. Unlike symmetry boundaries, **E** does not have to be tangential or normal to these boundaries. The only condition is that the fields on the two boundaries must have the same magnitude and direction (or the same magnitude and opposite directions).

Independent Boundary for an Electric Field Solution

The simulator computes the electric field on an independent boundary using the charges and voltages that you specified for the model as input. No other special conditions are imposed.

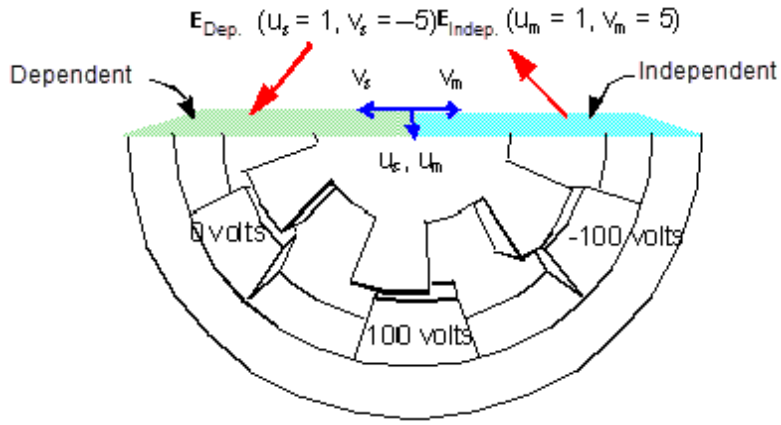
Dependent Boundary for an Electric Field Solution

The electric field on the dependent boundary is forced to match the field on the independent boundary. The magnitude of the electric field on both boundaries is the same. The fields on the two boundaries can either point in the same direction, or in opposite directions.

When to Use Matching Boundaries for an Electric Field Solution

Consider a simple electrostatic micromotor in which the rotor is held at zero volts and the six stator poles are switched between zero volts, 100 volts, and -100 volts. The E-field pattern at any point in time repeats itself every 180^{o} — causing the field in one half of the motor to match the field in the other half.

If you use matching boundaries, you only need to model half of the motor, as shown below. The E-field on the dependent boundary (the left side of the motor) is forced to match the magnitude and point in the opposite direction from the E-field on the independent boundary (the right side of the motor) - simulating the field pattern that would occur if the entire motor was modeled.



A symmetry boundary cannot be used in place of matching boundaries in this example. The electric field is not necessarily either perpendicular or tangential to the motor's periodic surfaces. In the example above, the electric field would be exactly tangential to the periodic surface only when the poles of the rotor are aligned with the poles of the stator. In the other positions of the rotor, the field is not tangential and matching boundaries are required.

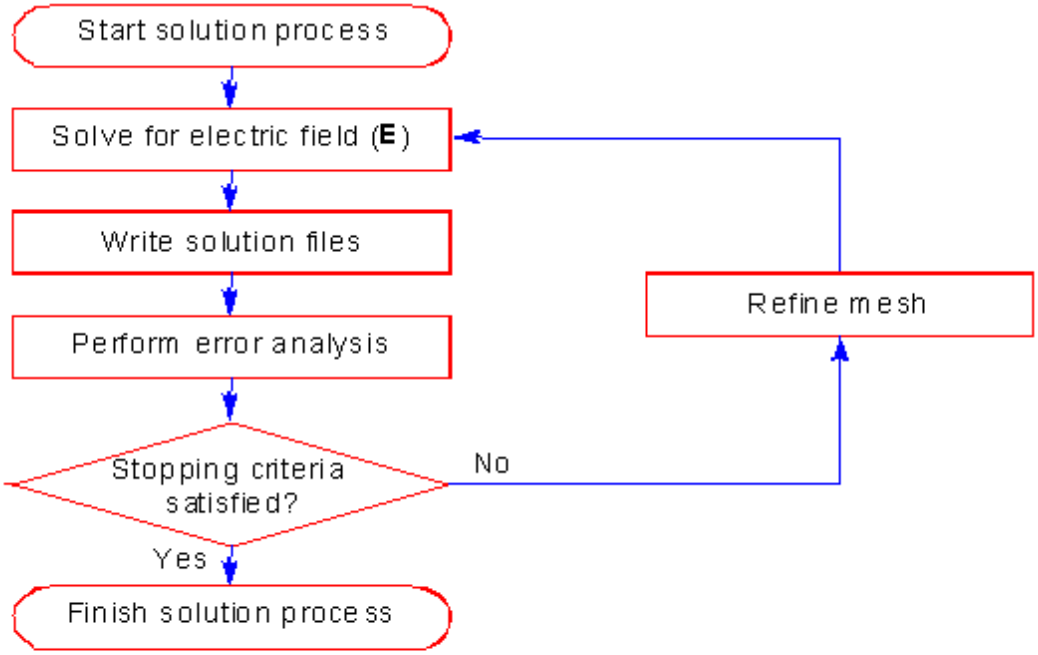
A **combined DC current flow + Electrostatic solution type** is also possible. In this case, the voltage distribution on conductors from the DC current flow solution is used as a Dirichlet (applied voltage) boundary condition in the electrostatic phase of the solution process. This type of sequence allows a certain class of problem involving electrostatic fields surrounding conductors with DC current flow to achieve a comprehensive solution. In this case, the solution provides the electric field in conductors and dielectrics.

Solution Process for an Electric Field Solution

Maxwell's electrostatic solver computes and stores the value of the electric potential at the vertices and midpoints of the edges of each tetrahedron in the finite element mesh. It solves for the electric field using the following relationship:

$$\vec{E} = -\nabla\Phi$$

After E is calculated, Maxwell writes out solution files and performs an error analysis. In an adaptive analysis, it refines the tetrahedra with the highest error, and continues solving until the stopping criterion is met. In a non-adaptive solution, the process stops.



Electric Field Energy for an Electric Field Solution

The energy, U, that is stored in a static electric field is given by the following equation:

$$U = \frac{1}{2}\int_{Vol} \vec{\mathbf{E}} \bullet \vec{\mathbf{D}} dVol$$

where:

- **E** is the electric field.
- **D** is the electric flux density.

Capacitance Matrix for an Electric Field Solution

Capacitance matrices can be expressed in terms of charge and voltage, or current and time-varying voltage. The unit of capacitance is F.

Note	For a capacitance matrix, the elements are defined based on excitations, not solids.

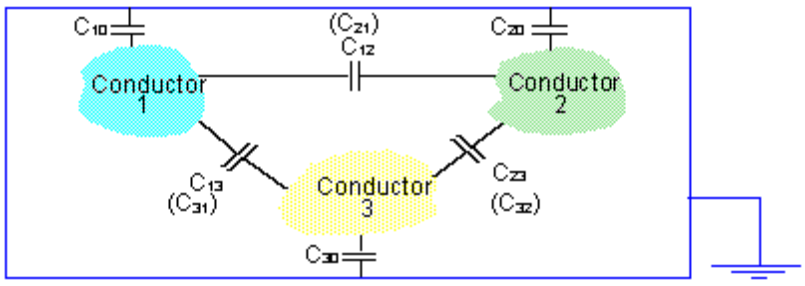
Capacitance in Terms of Charge and Voltage

A capacitance matrix represents the charge coupling within a group of conductors - that is, the relationship between their charges and voltages. Given a ground reference for the three conductors below, the net charge on each object is:

$$Q_1 = C_{10}V_1 + C_{12}(V_1 - V_2) + C_{13}(V_1 - V_3)$$

$$Q_2 = C_{20}V_2 + C_{21}(V_2 - V_1) + C_{23}(V_2 - V_3)$$

$$Q_3 = C_{30}V_3 + C_{31}(V_3 - V_1) + C_{32}(V_3 - V_2)$$



This can be expressed in matrix form as the following:

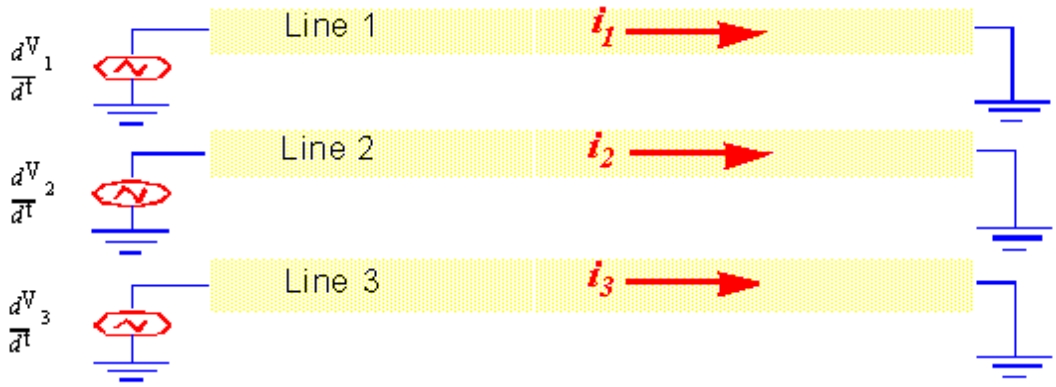
$$\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} = \begin{bmatrix} C_{10}+C_{12}+C_{13} & -C_{12} & -C_{13} \\ -C_{12} & C_{20}+C_{12}+C_{23} & -C_{23} \\ -C_{13} & -C_{23} & C_{30}+C_{13}+C_{23} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

The capacitance matrix above gives the relationship between Q and V for the three conductors and ground. An n-conductor device would have an n x n capacitance matrix.

Capacitance in Terms of Current and Time Varying Voltage

A capacitance matrix can also represent the relationship between currents and time varying voltages in a system of conductors. Given the three transmission lines shown below, the currents caused by the time varying voltage source on each line are given by the following equation:

$$\begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = \begin{bmatrix} C_{10}+C_{12}+C_{13} & -C_{12} & -C_{13} \\ -C_{12} & C_{20}+C_{12}+C_{23} & -C_{23} \\ -C_{13} & -C_{23} & C_{30}+C_{13}+C_{23} \end{bmatrix} \begin{bmatrix} \frac{dV_1}{dt} \\ \frac{dV_2}{dt} \\ \frac{dV_3}{dt} \end{bmatrix}$$



If dV_2/dt and dV_3/dt are set to zero, this relationship becomes the following:

$$\begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = [C] \begin{bmatrix} \frac{dV_1}{dt} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} C_{10} + C_{12} + C_{13} \\ -C_{12} \\ -C_{13} \end{bmatrix} \frac{dV_1}{dt}$$

This gives the currents that are induced on Line 2 and Line 3 when a time varying voltage source is applied to Line 1 - that is, the capacitive coupling between the three lines, or the short circuit capacitance.

The capacitance matrix can be exported to a circuit/system simulator, such as Twin Builder, where it can be used in a transient simulation.

Matrix Elements for an Electric Field Solution

If one volt is applied to Conductor 1 and zero volts are applied to the other two conductors, the capacitance matrix becomes the following:

$$\begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} = [C] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} C_{10} + C_{12} + C_{13} \\ -C_{12} \\ -C_{13} \end{bmatrix}$$

Diagonal Elements for an Electric Field Solution

These terms (such as C_{11}) are the sum of all capacitances from one conductor to all other conductors. These terms represent the self-capacitance of the conductors. Each is numerically equal to the charge on a conductor when one volt is applied to that conductor and the other conductors (including ground) are set to zero volts. For instance, $C_{11} = C_{10} + C_{12} + C_{13}$. This term is equal to the charge on Conductor 1 when it is set to one volt and the other conductors are set to zero volts.

Off-Diagonal Elements for a Capacitance Matrix

The terms in each column (such as C_{12} , C_{13}) are numerically equal to the charges induced on other conductors in the system when one volt is applied to a single conductor. For instance, in column one of the capacitance matrix shown above, C_{12} is equal to - C_{12}. This is equal to the charge induced on Conductor 2 when one volt is applied to Conductor 1 and zero volts are applied to Conductor 2.

The off-diagonal terms are simply the negative values of the capacitances between the corresponding conductors (the mutual capacitances). In column one of the example capacitance matrix, the off-diagonal terms represent the capacitances between Conductor 1 and the other two conductors; in column two, the terms represent the capacitance between Conductor 2 and the other conductors; and so forth.

Symmetry for a Capacitance Matrix

The capacitance matrix is symmetric about the diagonal. This indicates that the mutual effects between any two objects are identical. For instance, C_{13}, the capacitance between Conductor 1 and Conductor 3 (C_{13}), is equal to C_{31}, the capacitance between Conductor 3 and Conductor 1.

Solution Process for a Capacitance Matrix

To solve for the capacitance matrix, the electrostatic solver generates a single field solution. The system computes each entry of the capacitance matrix using the following relationship:

$$C = \int E \bullet D d\Omega$$

where the contributions of each conductor to the capacitance matrix are modeled using numerical matrix manipulation techniques.

$$D = \varepsilon E$$

Only conductors with voltage excitations can be included in a capacitance matrix calculation.

Lumped Capacitance

Lumped capacitance is essentially the result of a 1x1 capacitance matrix. It is used to calculate the electric field energy, U, from the E-field and D-field. Since the energy can also be expressed as:

$$U = \frac{1}{2} C V^2$$

the capacitance is:

$$C = \frac{2U}{V^2}$$

Maxwell assumes that the problem is set up so that the following is true:

- One conductor (or group of conductors) is set to +1 volt.
- Another conductor (or group of conductors) is set to zero volts.
- All other conductors are floating.

Therefore, the lumped capacitance between all conductors set to one volt and all those set to zero volts is simply 2U.

Warning	The lumped capacitance computation, as described above, is invalid if you assigned voltages other than +1V or 0V to the conductors in your model.

Note	For cases where a matrix extraction is desired in Maxwell, the respective extraction needs to be compatible with the hypothesis made regarding the equipotentiality of the respective surfaces, lack of radiation, etc., characteristic to a low frequency circuit

	approximation. In the circuit representation, the respective lumped elements are interconnected with other circuit elements at the nodes that physically represent the "terminals" in the electromagnetic field problem. Thus, it is possible (within the limits of the lumped circuit elements and applicable theory) to have circuit elements with field effects, such as capacitors and inductors, capable of storing energy. You can access these elements through the provided terminals/nodes, which can then be viewed as energy ports since most of the extracted elements have the capability to store energy. The volume charge density excitation is not a "terminal" type of excitation since the objects to which it is applied cannot, as a rule, be considered equipotential.

Conductance Matrix for a 3D DC Conduction Solution

The Maxwell 3D conductance matrix relates the currents and the voltages as:

$$\begin{bmatrix} I_1 \\ I_2 \\ \cdots \\ I_n \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} & \cdots & G_{1n} \\ G_{21} & G_{22} & ... & G_{2n} \\ ... & ... & ... & ... \\ G_{n1} & G_{n2} & ... & G_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ \cdots \\ V_n \end{bmatrix}$$

Or in more concise form:

$$I = GV$$

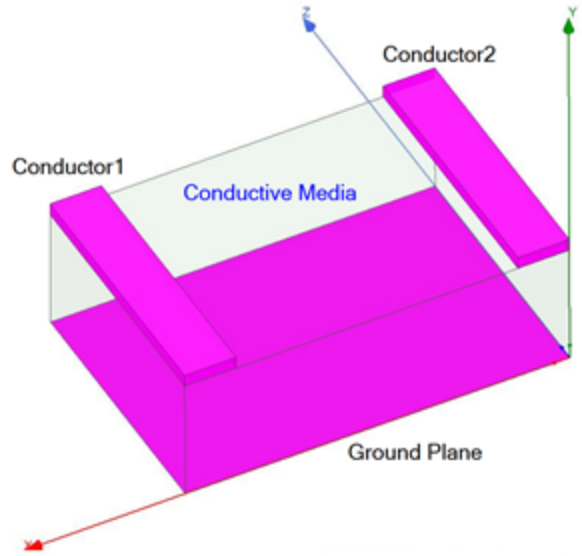
where:

- I is conduction current
- G is conductance
- V is voltage

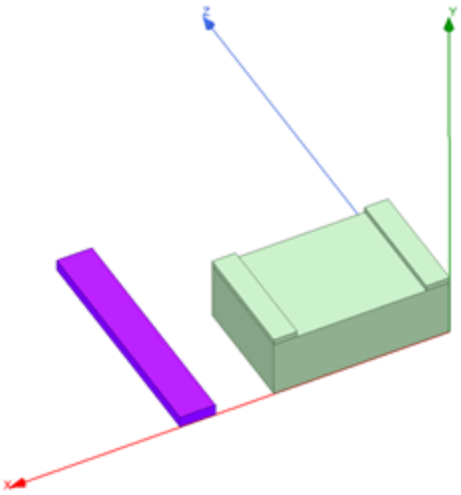
For a simple two conductor system with a ground plane (V=0), where there are conductive media in the middle. Referring to the figure below, the current flowing through **Conductor1** and **Conductor2**, I1 and I2, are related to their voltages V1 and V2 as:

$I_1=G_{11}*V_1+G_{12}*V_2$

$I_2=G_{21}*V_1+G_{22}*V_2$



Maxwell 3D can also handle designs that have disconnected conducting regions (multiple singly-connected regions).



The relationship between the Maxwell conductance matrix G^M and the Spice conductance matrix G^S is:

a. $G^M_{i,i} = G^S_{i,1} + G^S_{i,2} + G^S_{i,3} + \ldots + G^S_{i,n}$ (i=1,2,…,n)

b. $G^M_{i,j} = - G^S_{i,j}$ (for i not equal to j)

A ground reference is needed for extracting this matrix. For the Maxwell 3D DC conduction solver, the “sink” excitation, which can be assigned only to surfaces, will be treated as ground. If there is no sink excitation, you must explicitly choose at least one "ground" from the list of voltage excitations in the parameter matrix dialog box.

In the conductance matrix extraction, if there are n “signals” (which can be associated with 3D volume body or 2D sheet body), each one of the n signal ports will be excited with 1 volt while leaving the other signal ports at 0 volt (the ground will be kept at 0voltage). After calculating the solutions for the excitation settings, losses are computed using the entries of the Maxwell conductance matrix G^M.

Lorentz Force for an Electrostatic Calculation

The component of the Lorentz force due to current in an electric field is:

$$\boldsymbol{F} = \int_{Vol} q\boldsymbol{E}\, dVol$$

where:

- q is the charge.
- $\boldsymbol{E}$ is the electric field.

Warning	The Lorentz force does not compute the correct force on objects that are assigned materials with a relative permittivity greater than one. To find the force on these objects, use virtual force.

Lorentz Torque for an Electrostatic Calculation

The system uses Lorentz forces to compute the torque around the x, y, and z axes. The Lorentz torque is given by the following equation:

$$\bar{\boldsymbol{T}} = \int_{Vol} q(\bar{r} \times \bar{\boldsymbol{E}})dVol$$

where:

- $\boldsymbol{r}$ is the displacement vector from the axis of rotation.
- q is the charge.
- $\boldsymbol{E}$ is the electric field.

This integral is computed for the x-, y-, and z-axes, giving the net torque on the object(s) about each axis of rotation.

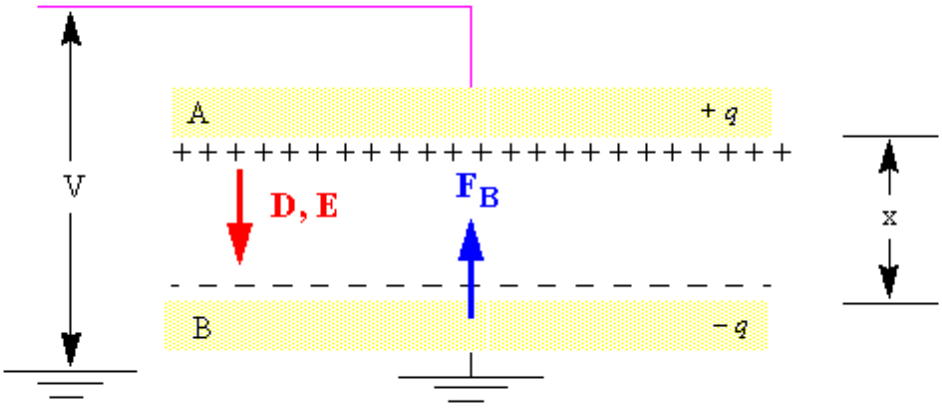
Warning	The Lorentz torque does not compute the correct torque on objects that are assigned materials with a relative permeability greater than one. To find the torque on these objects, use virtual torque.

Virtual Force for an Electrostatic Calculation

To compute the virtual force on an object, the electrostatic field simulator uses the principle of virtual work. In the structure shown below, the force on the bottom plate (plate B) in the direction of the displacement, x, is given by the following relationship:

$$\boldsymbol{F}_B = \left.\frac{dU(v,x)}{dx}\right|_{V = ConstantPotential}$$

where U is the electric field energy.



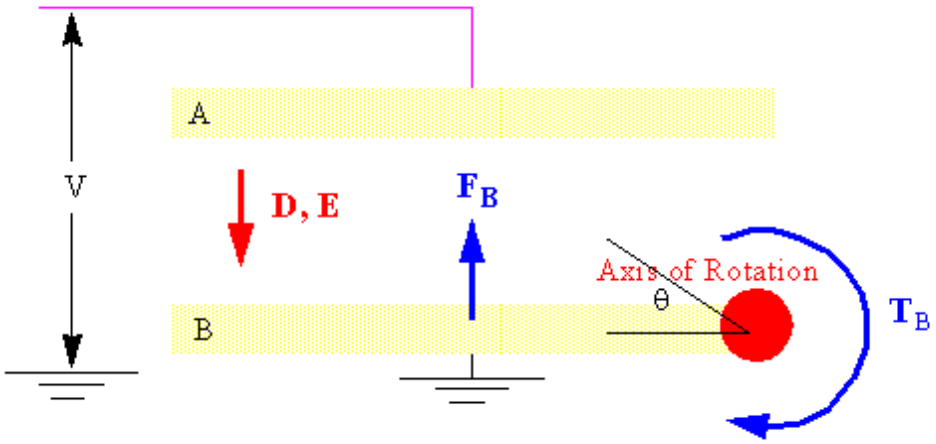
Unlike the classical virtual work method, the plate is not actually moved during the numerical process of the force computation. Instead, only the tetrahedra that lie along the outside surface of the object are virtually distorted. U and its derivative, *dU/dx*, are calculated from a single field solution using finite element interpolation functions.

Virtual Torque for an Electrostatic Calculation

Similar to the virtual force calculation, the system uses virtual work principles to compute the torque on an object. In the structure shown below, the virtual torque on the bottom plate (plate B) about the axis of rotation is given by the following relationship:

$$\boldsymbol{T}_B = \left.\frac{dU(v,\theta)}{d\theta}\right|_{V = ConstantPotential}$$

where U is the electric field energy.



Unlike the classical virtual work method, the plate is not actually rotated during the numerical process of the torque computation. Instead, only the tetrahedra that lie along the outside surface of the object are virtually distorted. U and its derivative, *dU/dθ*, are calculated from a single field solution using finite element interpolation functions.

Ohmic Loss for DC Conduction

The Ohmic loss, *P*, that is generated in conductors when solving DC current flow problems, is given by the following equation:

$$P = \int_{Vol} (\vec{E} \cdot \vec{J}) dVol$$

where

$\vec{E}$

- is the electric field.

$\vec{J}$

- is the electric current density.

For the sake of brevity, the ohmic loss density in the post processor should be labeled as **Ohmic Loss**. The total ohmic loss of an object or the whole arrangement can be calculated by integrating over a proper volume in the Field Calculator.

Surface Charge Density

The Surface Charge Density, ρ_s, that is generated on a surface between two materials when solving electric field problems, is given by the following equation:

$$n \cdot (D_2 - D_1) = \rho_s$$

$$n \cdot (\varepsilon_2 E_2 - \varepsilon_1 E_1) = \rho_s$$

where

- E_1 is the electric field and ε_1 the permittivity in the material on one side of the boundary.
- E_2 is the electric field and ε_2 the permittivity in the material on the opposite side of the boundary.
- n in the normal vector of the surface.

To calculate the surface charge, you must first select a surface in the modeling window.

Surface Current Density

For impedance boundary, the Surface Current Density, *Jsurf*, is computed as the cross product of the surface normal of the impedance boundary and the difference between H fields on the top and bottom of the boundary., and is given by the following equation:

$$Jsurf = n \times (H_{top} - H_{bottom})$$

where

- n is the surface normal vector.
- $(H_{top} - H_{bottom})$ is the difference between H fields on the top and bottom of the impedance boundary.

In order to calculate the surface current density, you must first select a surface in the modeling window.

Magnetostatic Field Calculation

In a **magnetostatic solution**, the magnetic field is produced by DC currents flowing in conductors/coils and by permanent magnets. The electric field is restricted to the objects modeled as real (non-ideal) conductors. The electric field existing inside the conductors as a consequence of the DC current flow is totally decoupled from the magnetic field. Thus, as far as magnetic material properties are concerned, the distribution of the magnetic field is influenced by the spatial distribution of the permeability. There are no time variation effects included in a magnetostatic solution, and objects are considered to be stationary. The energy transformation occurring in connection with a magnetostatic solution is only due to the ohmic losses associated with the currents flowing in real conductors.

The magnetostatic field solution verifies the following two Maxwell's equations:

$$\nabla \times \vec{H} = \vec{J}$$

$$\nabla \cdot \vec{B} = 0$$

with the following constitutive (material) relationship being also applicable:

$$\vec{B} = \mu_0(\vec{H} + \vec{M}) = \mu_0 \cdot \mu_r \cdot \vec{H} + \mu_0 \cdot \vec{M}_p$$

where:

$\vec{H}(x,y,z)$

- is the magnetic field strength.

$\vec{B}(x,y,z)$

- is the magnetic flux density.

$\vec{J}(x,y,z)$

- is the conduction current density.

$\vec{M}_p(x,y,z)$

- is the permanent magnetization.

$\mu_0 = 4 \cdot \pi \cdot 10^{-7}\ H/m$

- is the permeability of vacuum.

μ_r

- is the relative permeability.

For nonlinear materials, the dependency between the H and B fields is nonlinear and can be isotropic or orthotropic (in the case of anisotropic material behavior, μ_r is a tensor). Similarly, for permanent magnets, nonlinearity can occur in practical cases and is allowed. Additionally, if a demagnetized condition is to be taken into account for (nonlinear) permanent magnets operating below the knee, Maxwell provides an advanced setup option allowing a solution based on a previously computed demagnetization operating point. If nonlinearity occurs in soft materials (with negligible hysteresis) simultaneously with orthotropic behavior, Maxwell requires that the BH curves for the principal directions in the respective material(s) be provided. From these curves, the energy dependency on H is extracted for each of the respective principal directions and is used in the process of obtaining the nonlinear permeability tensor used in the Newton-Raphson iterative solution process:

$$\vec{B} = \vec{B}_0 + [\tilde{\mu}] \cdot (\vec{H} - \vec{H}_0)$$

$$[\tilde{\mu}] = \frac{\partial \vec{B}}{\partial \vec{H}} = [\Delta\tilde{\mu}] + [\mu]$$

where $\vec{B}_0$ and $\vec{H}_0$ are the previous field solution, $[\Delta\tilde{\mu}]$ is a general full tensor, and $[\mu]$ is given by the following:

$$[\mu] = \begin{bmatrix} \mu_x & & \\ & \mu_y & \\ & & \mu_z \end{bmatrix}$$

μ_x, μ_y, μ_z with taking into account the anisotropic effects of any laminations present in the model.

The 3D magnetostatic solver considers the magnetic field H with the following components:

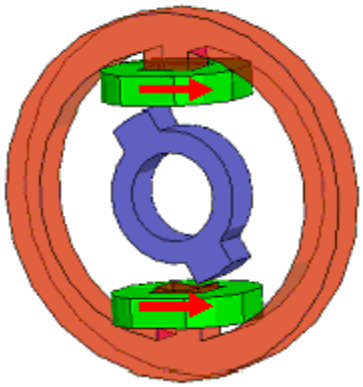
$$\vec{H} = \vec{H}_p + \nabla\varphi + \vec{H}_c$$

where φ is the magnetic scalar potential, $\vec{H}_p$ is a particular solution constructed by assigning values to all the edges in the mesh in such a way that Ampere's law holds on all contours of all tetrahedra faces in the mesh, and $\vec{H}_c$ accounts for the permanent magnets if any. Thus, the DOFs are the nodal values of the magnetic scalar potential with ten values per tetrahedron at each of the four vertices and all six mid edge nodes, ensuring a quadratic approximation inside each finite element.

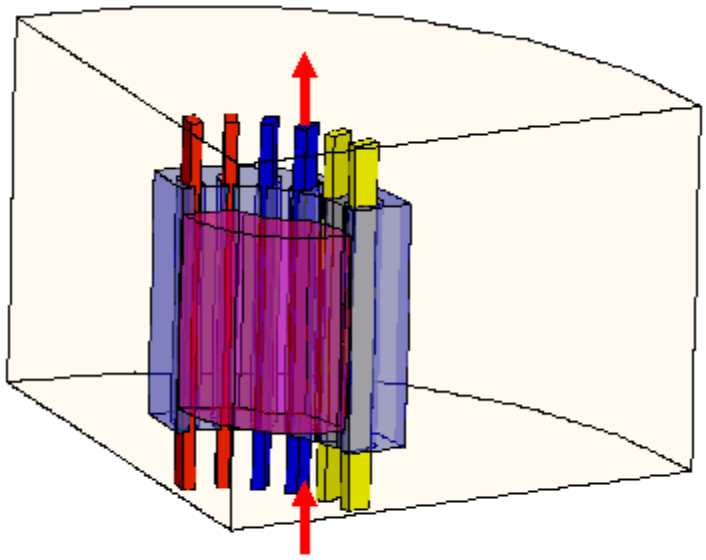
There are major advantages of this formulation over other existing ones, including using considerably fewer computational resources (due to the scalar nature of the DOFs), not requiring a gauge due to excellent numerical stability, significantly reducing cancellation errors, and capably of automatically multiplying connected iron regions. The magnetostatic solver handles both 3D linear and nonlinear problems. In the case of nonlinear applications, a classic Newton-Raphson iterative algorithm with user-controlled accuracy is used.

The magnetostatic solver calculates the magnetic field distribution produced by a combination of known DC current density vector distribution and a spatial distribution of objects with permanents magnetization. It is also possible to apply boundary conditions to a model such that the simulation of the immersion of a device into an external magnetic field is also possible. In this latter case, the boundary conditions should be applied in such a way that Maxwell's equations are not violated inside the domain of the solution or at the boundaries.

Typical sources for magnetostatic field problems include voltage, current, permanent magnetization, and current density. When applying the sources for the magnetic field problems, the applied current distribution must be divergence free in the entire space of the solution as it is physical for quasi-stationary conduction current density distributions. Thus, the conduction paths (s) for the applied current distributions must be closed when totally contained within the solution space for the problem, or must begin and end at the boundaries.



Conduction paths in the two coils are closed (one current excitation applied on terminals in each coil)



Current excitations are applied at the boundaries at both ends of conductors (no "terminals" need to be defined)

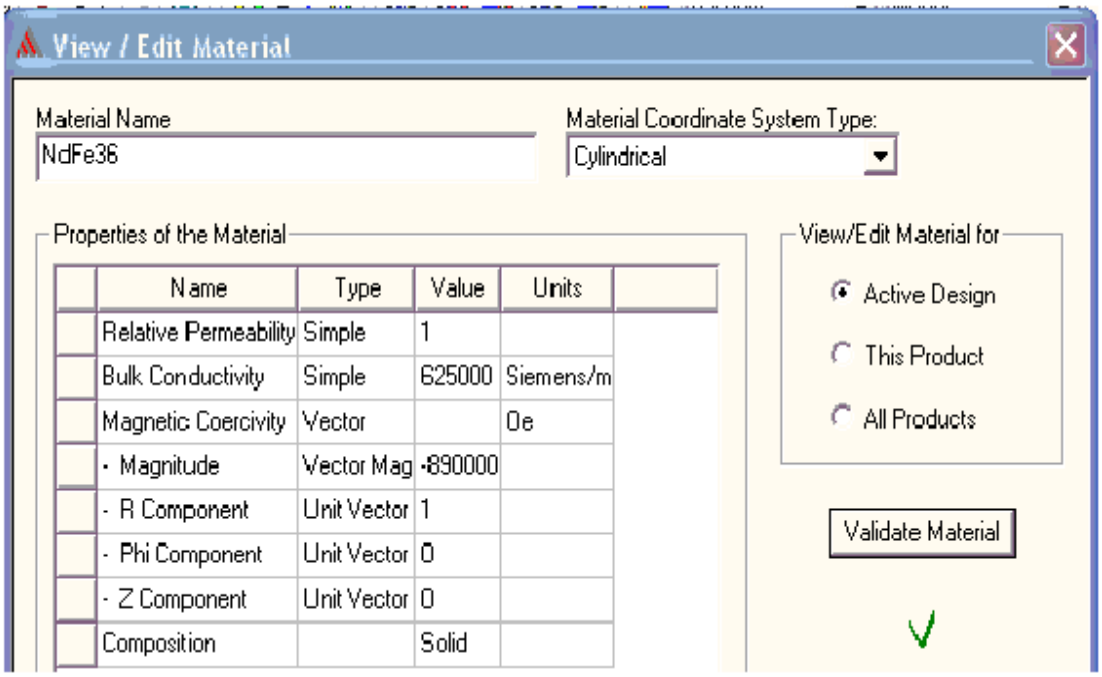
The total current applied to conductors that touch the boundaries does not require the existence of terminals at the ends where the current is applied because the respective planar surfaces of the conductors in the plane of the region (background) can be used to apply the excitations. In the case of a closed conduction path, a terminal (sheet object) must be created, matching the cross-section of the conductor at the desired location such that the excitation (current/voltage) can be applied.

Note	Current and voltage excitations cannot be mixed on any conduction path.

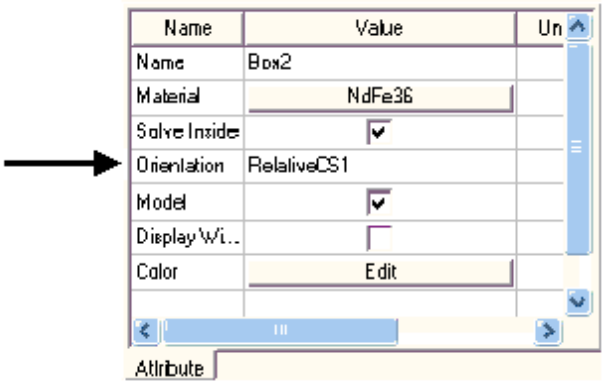
Voltage excitations (at least two) must be specified such that the current can flow in the respective conductors to form an uninterrupted conduction path. **Voltage drop** excitation should by used in the case of closed conduction paths (possibly containing objects with different electric conductivities), primarily in situations where the total current is unknown. One voltage drop excitation per conduction path should be applied. The total current for any conduction path should add up to zero.

The magnetostatic solver does not compute the electric field distribution associated with a voltage distribution outside conductors. However, the electric field inside conductors is indirectly available for post processing via the conduction current density distribution, which is available. Thus it is possible to calculate the ohmic loss in conductors with applied current or voltage excitations.

Permanent magnetization in Maxwell® is treated as a material property, but we mention it here because of its source characteristics.



Permanent magnetization can be specified in Cartesian, cylindrical, or spherical material coordinate systems, as shown in the above window where the R, Phi, and Z components (in a cylindrical system, considered here as example) can be specified in any meaningful combination. The actual direction of the vector property is decided when the property is assigned to the object and when the desired coordinate system is chosen (the global coordinate system always exists, while additional ones can be defined). If necessary, additional relative coordinate systems can be defined and used to specify the orientation of the vector property.



While the type of material coordinate system is embedded into the material definition, the actual orientation is not, such that a material property defined, for instance, in a cylindrical system can be applied to multiple objects with different orientations if multiple relative local coordinate systems were defined and used in the process of assigning the respective material property.

A **typical boundary conditions** used in magnetostatic problems is the **magnetic field tangent** (the default, natural boundary condition that is automatically applied on all surfaces of the problem space -the surfaces of the geometry entity containing the model inside it). This default boundary condition can be overwritten if other boundary conditions are applied on exterior surfaces of the solution space. The default boundary condition confines the magnetic field to the solution space; therefore, this boundary must be placed at some distance from the sources to avoid over-constraining the fields by placing the boundaries to close to the model objects. While it is difficult to provide "recipes" with general validity on the placement of the boundaries of problems, a good rule of thumb asserts that if a model can be imagined as being contained in a sphere of radius R, then the boundaries can be placed at a 4-5 radii R from the imaginary center of the model.

The **Zero Tangential H Field** allows the user to prescribe a normal (in average) field orientation on an arbitrary surface. This boundary condition does not require any further user input, meaning that no value and coordinate system specification are necessary.

In the case of **Tangential H Field** boundary condition, the values of two tangential components and a surface coordinate system must be defined and subsequently used in the assignment process of this boundary condition. This boundary condition is restricted to two kinds of surfaces: planar or cylindrical. The CS in the planar case would be a rectangular coordinate system with X and Y components of H in the plane. The coordinate system in the cylindrical case would be a coordinate system with a Z axis coinciding with the axis of the cylinder, and the two components to input would be the PHI and Z axis components of H.

In the case of the tangential H field, caution is advised: This boundary condition should be specified such that Ampere's theorem is not violated inside the field domain or at the boundaries.

Symmetry boundary conditions are used to solve problems with symmetry and, thus, allow the users to take advantage of the significantly reduced problem size for a given accuracy.

Symmetry boundary conditions can be one of two kinds:

- Odd (flux tangent)
- Even (Flux normal)

Although the use of this symmetry boundary condition overlaps with previous conditions to some extent, the symmetry boundary conditions should only be used in clear symmetry cases.

Symmetry for a Magnetostatic Field Solution

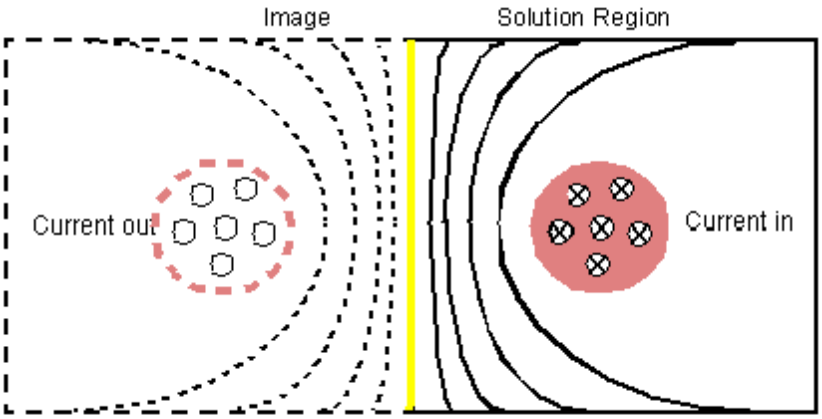
A Symmetry boundary models a plane of symmetry in a structure. Use this type of boundary condition to take advantage of geometric and magnetic symmetry in a structure. Doing so enables you to reduce the size of your model, which helps to conserve computing resources. Two types of symmetry are available:

- Odd Symmetry (Flux Tangential)
- Even Symmetry (Flux Normal)

These boundaries can only be assigned to the outside edges of the solution region.

Odd Symmetry (Flux Tangential) for a Magnetostatic Field Solution

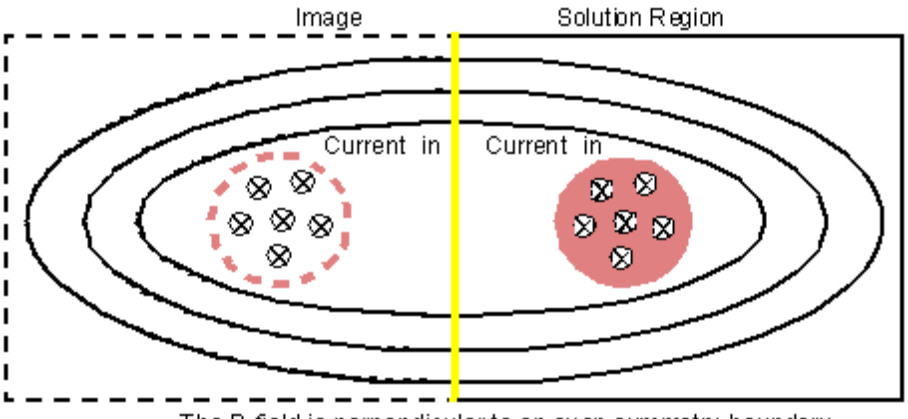
Use an odd symmetry boundary to model a plane of symmetry in which current on one side of a plane flows in the opposite direction to current on the other side of the plane. Magnetic flux is tangential to this type of boundary. To define an odd symmetry boundary, the simulator sets the selected edge to a Neumann boundary.



The B-field is tangential to an odd symmetry boundary.

Even Symmetry (Flux Normal) for a Magnetostatic Field Solution

Use an even symmetry boundary to define a plane of symmetry where the direction of current flow is the same on both sides of the plane. Magnetic flux is normal to this type of boundary. To define an even symmetry boundary, the simulator sets the selected edge to a Magnetic Field boundary with a field value of zero - acting as a magnetic mirror to the model.



The B-field is perpendicular to an even symmetry boundary.

Warning	When using even symmetry boundaries, be careful that you do not violate Ampere's

	law.

Matching Boundary for a Magnetostatic Field Solution

Matching boundaries allow you to model planes of periodicity where the H-field on one surface exactly matches the H-field on another. They force the magnetic field at each point on one surface (the "independent" boundary) to match the magnetic field at each corresponding point on the other surface (the "independent" boundary). They are very useful for modeling devices such as motors, in which the magnetic field repeats every 180^o, 120^o, 90^o, or less. Basically, they enable you to model the smallest possible periodic segment of the device - reducing the amount of computing resources needed during the solution.

To set up matching boundaries, you must create both an independent boundary and a dependent boundary. Unlike Symmetry boundaries, **H** does not have to be tangential or normal to these boundaries. The only condition is that the fields on the two boundaries must have the same magnitude and direction (or the same magnitude and opposite directions).

Independent Boundary for a Magnetostatic Field Solution

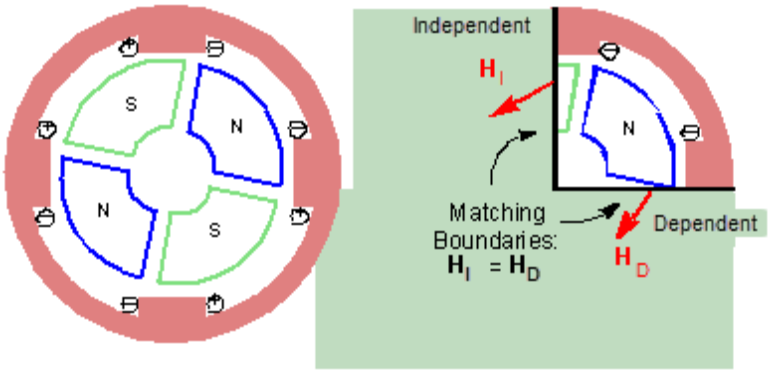
The simulator computes the magnetic field on an independent boundary using currents, permanent magnets, and magnetic fields as input. The field is then mapped to the dependent boundary.

Dependent Boundary for a Magnetostatic Field Solution

The magnetic field on the dependent boundary is forced to match the field on the independent boundary. The magnitude of the magnetic field on both boundaries is the same. However, the fields on the two boundaries can either point in the same direction, or in opposite directions, as specified by the user during setup.

When to Use Matching Boundaries for a Magnetostatic Field Solution

Matching boundaries enable you to take advantage of periodicity in a structure. For example, below is a diagram of the cross-section of a simple brushless DC motor. The field in such a motor repeats itself every 90 degrees; that is, the field pattern in one quarter of the motor matches the magnitude and direction (or the opposite of the direction) of the field pattern in the other three quarters. With matching boundaries, all you have to model is one quarter of the structure.

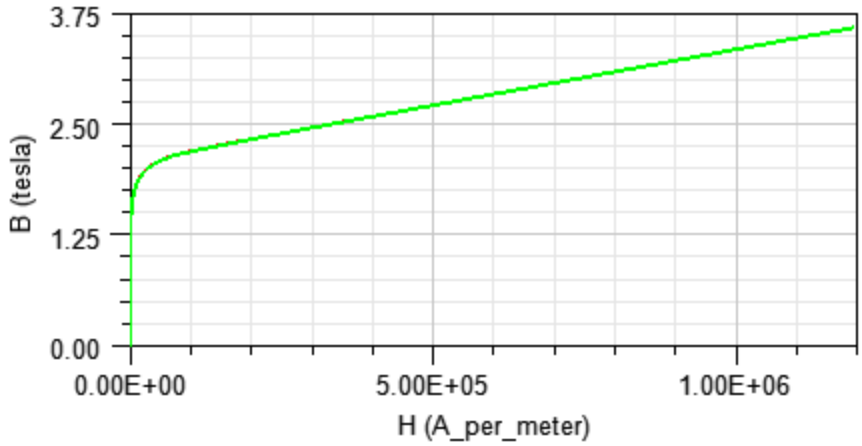


A Symmetry boundary cannot be used to simulate periodicity because the magnetic field is not necessarily either perpendicular or tangential to periodic surfaces. For example, in the quarter model shown on the right, the magnetic field is exactly perpendicular to the bounding surfaces only when the gap separating the permanent magnets is perfectly horizontal or vertical. For all other positions of the rotor, matching boundaries are required.

Nonlinear Materials

Magnetostatic

If a material has a permeability that varies with the flux density, a **B** vs. **H** curve (B-H curve) such as the one below is needed to describe the material's nonlinear behavior.



In nonlinear materials, the B-field (magnetic flux density) is a function of itself:

$$\boldsymbol{B} = \mu_r(B)\mu_o H$$

where $\mu_r(B)$, the relative permeability, depends on the magnitude of the B-field at each point in the material. Therefore, to model the magnetic behavior of the material, a curve relating the B-field directly to the H-field is used to describe the nonlinear relationship.

See also Advanced Nonlinear Options for Magnetostatic Solutions or Magnetization for Transient Solutions for more information about permeability and magnetization settings.

Solution Process for a Magnetostatic Field Solution

In general, the solution process involves a phase in which the distribution of current density (if not already known) is obtained, followed by the calculation of the distribution of magnetic field that corresponds to the distribution of the sources. In the case of solid conductors with an applied current source, an automatic process is initiated such that following an iterative process involving adaptive mesh refinement, the current density vector distribution is obtained and subsequently the corresponding magnetic field distribution is computed. In the case of stranded conductors, it is assumed that the current density is uniform on the cross-section of the respective conductors. Thus, the stranded option for current excitations should mostly be used only in cases where the cross-section of conductors is constant, consistently with the assumption that the respective object is a coil built with strands of wire. For this reason, conductors (coils) could be created by either sweeping around an axis using zero segments, or sweeping along a path having only true surface segments in order to create a smooth surface coil having a completely uniform cross-sectional area over its entire length.

Magnetic Field Energy for a Magnetostatic Field Solution

The magnetic energy of a system is given by the following expressions. The expressions represent total values of energy for the volumes taken into account. Note that the integrals have simpler expressions if the material property of the object considered is a linear one. In case of linear material properties magnetic energy and co-energy values are identical (W = Wc).

- In linear materials, the energy is:

$$W = \frac{1}{2} \int_{Vol} \boldsymbol{B} \bullet \boldsymbol{H} \, dVol$$

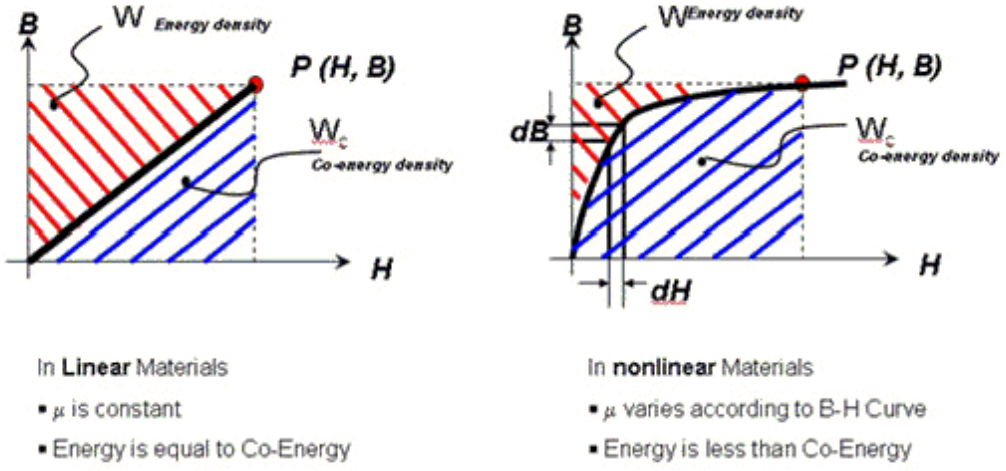
- In nonlinear materials, the energy is:

$$W = \int_{Vol} w \bullet dVol = \int_{Vol} \left(\int_{0}^{B} (\boldsymbol{H} \bullet d\boldsymbol{B}) \right) dVol$$

where:

- **H** is the magnetic field.
- **B** is the magnetic flux density.

The following figure represents graphically the definitions of both energy and co-energy densities. Note that both are local quantities (i.e. are function of X, Y, Z coordinates of the location in the model) and also depend on the operating point on the applicable B-H curve. Energy is equal to the area above the BH-curve, and coenergy is equal to the shaded blue area below the BH-curve.



For the sake of brevity, the magnetic flux density in the post processor should be labeled as **Magnetic Flux**. The total magnetic flux of an object or the whole arrangement can be calculated by integrating over a proper volume in the Field Calculator.

Related Topics

Technical Notes: Magnetic Co-energy

Technical Notes: Magnetic Apparent Energy

Magnetic Coenergy for a Magnetostatic Field Solution

The magnetic coenergy of a system is given by the following expressions. The expressions represent total values of co-energy for the volumes taken into account. Note that the integrals have simpler expressions if the material property of the object considered is a linear one. In case of linear material properties magnetic energy and co-energy values are identical (W = Wc).

- In linear materials, the coenergy is:

$$W_c = \frac{1}{2}\int_{Vol} \boldsymbol{B} \bullet \boldsymbol{H}\, dVol$$

- In nonlinear materials, the coenergy is:

$$W_c = \int_{Vol} w_c \bullet dVol = \int_{Vol}\left(\int_0^H (\boldsymbol{B} \bullet d\boldsymbol{H})\right) dVol$$

where:

- **H** is the magnetic field.
- **B** is the magnetic flux density.

The coenergy is related to the magnetic field energy.

Note	In case permanent magnets are part of the model, as a rule, the magnetic energy reported by the solver and the post processor may be different. This occurs because the adaptive meshing part of the solution sequence runs a special computation so that it can avoid situations with a total zero energy of the field (problems with permanent magnets and no other excitation). To avoid such situations, the solver uses (and reports) the total energy, which includes the absolute value of the energy inside the permanent magnets. Because of this, the energy reported by the solver in these cases is greater than the energy reported by the post processor.

Related Topics

Technical Notes: Magnetic Field Energy

Technical Notes: Magnetic Apparent Energy

Magnetic Apparent Energy for a Magnetostatic Field Solution

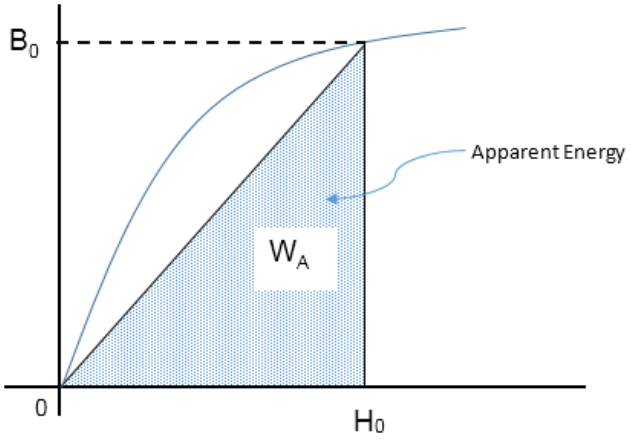
The magnetic apparent energy of a system is given by the following expression:

$$W_A = \frac{1}{2} H_0 B_0$$

where:

- $\mathbf{H_0}$ is the magnetic field final operating point.
- $\mathbf{B_0}$ is the magnetic flux density final operating point .

Apparent energy is equal to the shaded area below.



Related Topics

Technical Notes: Magnetic Field Energy

Technical Notes: Magnetic Co-Energy

Inductance Matrix for a Magnetostatic Field Solution

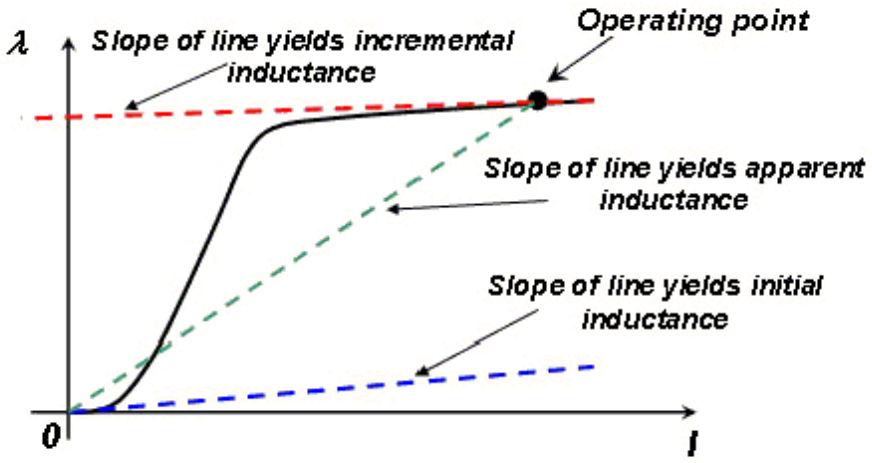
Inductance matrices can be expressed in terms of flux linkage and current, or in terms of voltage and time-varying current. Inductance values are given in henries.

Types of Inductance

For nonlinear materials, the definition of inductance requires additional detail compared to the linear case. Three commonly used inductance values are illustrated in the figure above - each inductance has useful applications. Maxwell uses apparent inductance to calculate flux linkage as a function of the independent variables, since it changes with current as the material properties change. To obtain apparent inductances for nonlinear materials in Maxwell, a two-step procedure is followed:

1. A nonlinear magnetostatic solution is generated with all sources at user specified values. This establishes a value of permeability that varies with each mesh element, since the degree of saturation varies throughout the device.
2. These self-consistent values are used in a linear solution for the inductance matrix, with each coil current set to one ampere. The resulting values are apparent inductances, which vary with each specified coil current operating point because the material properties change.

The inductance calculation is now linearized on the self-consistent values. If the coil currents had been increased to the specified values in the first step, in the linearized model, the resulting magnetic field would be identical to the nonlinear solution. Therefore, the inductances from the second step are the ratios of respective flux linkages to coil current, at the nonlinear operating point originally determined.



Nonlinear Inductance — Current vs. Flux Linkage

The slope of any line drawn on the figure above has units of inductance (henries) and any area enclosed on this plot represents units of energy in joules if magnetic flux is expressed in Webers and current is expressed in Amperes. The three inductances are defined as:

Incremental (differential)	The slope of a line tangent to the curve at the operating point. Incremental inductance is the usual textbook definition. For small signal AC analysis in

inductance $L_{inc} = \frac{d\lambda}{dI}$	Maxwell SPICE or Saber, incremental inductance is evaluated using an operating point determined from a DC solution. Incremental inductance can also be used in time domain system simulations by using the compensation theorem, but it has no useful relationship to stored energy.
Apparent inductance $L_{app} = \frac{\lambda}{I}$	The slope of a line from the origin to the operating point. Apparent inductance gives the total flux linkage as a function of current, so it is well-suited for the state equation or basis function approach to time domain simulations. For small-signal AC analysis, apparent inductance cannot be used directly, but can be used to supply the proper derivatives to Maxwell SPICE. Energy can be calculated with apparent inductance, but the result is not really the stored energy because it ignores the path taken along the nonlinear curve. Apparent inductance does not provide the actual stored energy at the specified operating point. However, during time domain simulations, apparent inductance varies with current, and tracks the nonlinear curve shown in the figure.
Initial inductance L_{init}	The slope of a line tangent to the curve at the origin. Initial inductance applies to many magnetic materials that have a "toe" in the magnetization curve. Laboratory measurements at 0 A DC measure initial inductance, not the "linear" value that may be somewhat higher. Note: Initial inductance is a particular case of the incremental inductance where the slope of the respective line passes through the origin of the (flux - current) plane.

For linear materials, all three inductance values are equal. For nonlinear materials, the three values are generally different. Except for initial inductance, each value varies with the operating point. Ignoring the "toe" of the curve, the relation Lapp > Linc holds. Energy taken from or supplied to the external circuit, as determined from the terminal voltage and current, will be correct.

The apparent inductance calculated by Maxwell at the actual operating point due to all sources in the model (currents in the coils but also permanent magnets) is the base of all inductance calculations. Incremental (differential) inductance information can be easily derived from the flux vs current characteristic.

It should be noted that the apparent value relates to the local frozen, current dependent absolute permeability as described above in the two step process. Incremental inductance relates to another physical quantity which is the differential (incremental) permeability defined as dB/dH.

Solver calculated magnetostatic energy (reported as part of the solution convergence data) is the apparent energy, always equal to the global, model-wise average between energy and co-energy. For the linear material case in the entire model there is only one value of the inductance L = L*init* = L*incr* = L*app*.

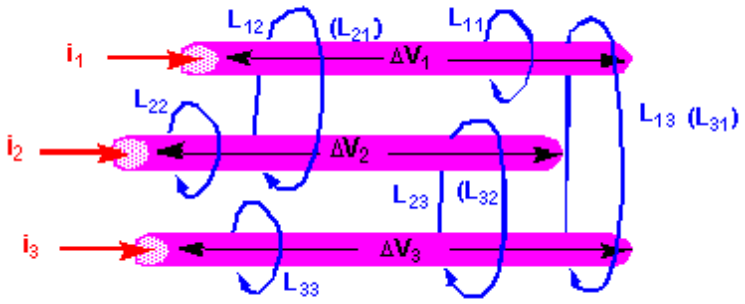
Inductance in Terms of Flux Linkage and Current

An inductance matrix represents the magnetic flux linkage between current loops. Given the three current loops below, the relationship between induced flux and currents is:

$$\lambda_1 = L_{11} i_1 + L_{12} i_2 + L_{13} i_3$$

$$\lambda_2 = L_{21} i_1 + L_{22} i_2 + L_{23} i_3$$

$$\lambda_3 = L_{31} i_1 + L_{32} i_2 + L_{33} i_3$$



This can be expressed in matrix form as:

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix}$$

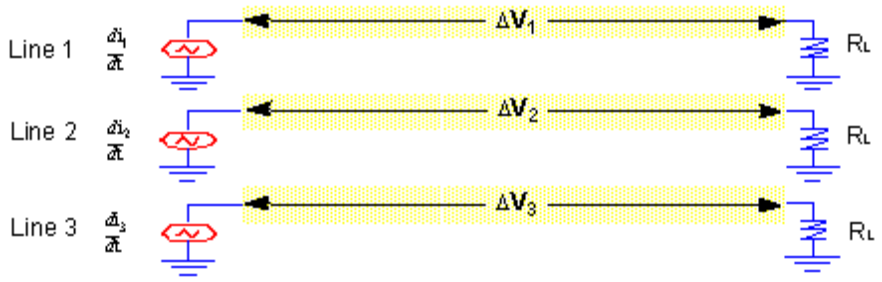
λ

The inductance matrix above gives the relationship between and i for the three independent current loops. A device with n current loops would have an n n inductance matrix.

Inductance in Terms of Voltage and Time-Varying Current

An inductance matrix can also represent the relationship between voltage and current fluctuations in a system. Given the three independent transmission lines shown below, the voltage changes caused by the time varying current source on each line are given by:

$$\begin{bmatrix} \Delta V_1 \\ \Delta V_2 \\ \Delta V_3 \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} & L_{13} \\ L_{21} & L_{22} & L_{23} \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \begin{bmatrix} \frac{di_1}{dt} \\ \frac{di_2}{dt} \\ \frac{di_3}{dt} \end{bmatrix}$$



Δ

The inductance matrix above gives the relationship between V and dI/dt for the three transmission lines. If di_2/dt and di_3/dt are set to zero, this relationship becomes:

$$\begin{bmatrix} \Delta V_1 \\ \Delta V_2 \\ \Delta V_3 \end{bmatrix} = [L] \begin{bmatrix} \frac{di_1}{dt} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} L_{11} \\ L_{12} \\ L_{13} \end{bmatrix} \frac{di_1}{dt}$$

This gives the voltage changes that are induced on Lines 2 and 3 when a time-varying current source is applied to Line 1 - that is, the inductive coupling between the three loops.

The inductance matrix can be exported to a circuit/system simulator, such as Twin Builder, where it can be used in a transient simulation.

Matrix Elements for an Inductance Matrix

If one amp is applied to Current Loop 1 and zero amps is applied to the other two loops, the inductance matrix becomes the following:

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = [L] \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} L_{11} \\ L_{12} \\ L_{13} \end{bmatrix}$$

Diagonal Elements for an Inductance Matrix

Terms such as L_{11} represent the self-inductance of each current loop. Self-inductance is numerically equal to the flux linkage in a current loop when one amp is flowing in it, and no current is flowing in the other loops. For example, L_{11} is equal to the flux in Current Loop 1 when one amp is flowing in that current loop, and no current is flowing in the other loops.

Off-Diagonal Elements for an Inductance Matrix

Terms such as L_{12} and L_{13} represent the mutual inductances between the current loops. Mutual inductance is numerically equal to the flux linkage in a current loop when one amp is flowing through another loop, and no current is flowing anywhere else. For example, L_{12} is equal to the flux linkage in Loop 1 when one amp is applied to Loop 2 and no current is flowing in the other loops.

Symmetry for an Inductance Matrix

The inductance matrix is symmetric about the diagonal. This indicates that the mutual effects between any two loops are identical. For instance, L_{13}, the inductance between Current Loop 1 and Current Loop 3, is equal to the inductance between Current Loop 3 and Current Loop 1.

Solution Process for an Inductance Matrix

To solve for the inductance matrix, the magnetostatic solver generates a single field solution. The system computes each entry of the inductance matrix using the following relationships:

$$U = \frac{1}{2}LI^2 = \frac{1}{2}\int H \bullet B d\Omega$$

But, assuming that the current, i, is one ampere:

$$L = \int H \bullet B d\Omega$$

where:

$$B = \mu H$$

Each conductor in the matrix is assumed to be part of an independent current loop. The contributions of each loop to the inductance matrix are modeled using numerical matrix manipulation techniques.

Warning	Inductance is valid for different current levels in linear and nonlinear models: • In models containing only linear materials, the system assumes that one amp of current flows through each loop. The inductance matrix that is computed for these models is valid for all current levels. • In models containing nonlinear materials, the system uses the currents you specified when setting up boundaries and sources. The inductance matrix that is computed for these models is valid only for these current levels.

Grouping Inductance Matrix Elements

The results of the inductance matrix calculation may be post processed into series/parallel windings using the grouping function. The operations of the grouping function can be one of two cases:

- Grouping all coils in series connection by grouping into 1 branch
- Grouping in a mixed series/parallel combination by setting the number of branches to a value of 2 or more.

Series Grouping

The inductances of the coils can be calculated as follows:

$$Lcoil_i = L_i \cdot n_i^2$$

Where $\mathbf{L_i}$ is the per-turn inductance of the source calculated by the field solver and $\mathbf{n_i}$ is the number of turns specified in the grouping dialog box. Combining inductors in series, which is the case when the number of branches is set to 1, is done according to:

$$L_{seriesgroup} = \sum_{i=1}^{m} Lcoil_i + \sum_{\substack{i,j=1 \\ i \neq j}}^{m} M_{ij} \cdot n_i \cdot n_j + \sum_{\substack{i,j=1 \\ i \neq j}}^{m} M_{ji} \cdot n_i \cdot n_j$$

where:

- *m* is the number of source entries being grouped.
- $Lcoil_i$ represents the self inductance term for each coil entry to be grouped according to the equation above.
- *Mij* and *Mji* are the mutual inductance terms between the sources being grouped. *Mij* and *Mji* must generally be considered separately as the matrix may be unsymmetric.
- n_i are the number of turns for each coil in the group.

Parallel Branches

If the number branches in the group is set to a value other than 1, the coils are first grouped in series as described above, and then divided into by the number if branches specified.

The grouped inductance is then calculated as follows:

$$L_{branch} = \frac{L_{seriesgroup}}{k^2}$$

Where *k* is the number of branches set for the group.

The calculation above only relies on the inductance matrix calculated during the field solution and can therefore be manipulated as a post processing calculation.

Lorentz Force

The component of the Lorentz force due to current in a magnetic field is:

$$\boldsymbol{F} = \int_{Vol} \boldsymbol{J} \times \boldsymbol{B}\, dVol$$

where:

- ***J*** is the current density.
- ***B*** is the magnetic flux density.

Note	The Lorentz force does not compute the correct force on objects that are assigned materials with a relative permeability greater than one (that is, materials where $\mu \neq \mu_0$). To find the force on these objects, use virtual force.

Note: For PCB applications, the parameter force calculated by using virtual force is not accurate if the mesh in air gaps between two layers is not fine enough. It is strongly recommended to compute parameter forces on conducting objects by using the Lorentz force option.

Lorentz Torque

The system uses Lorentz forces to compute the torque around the x-, y-, and z-axes. The Lorentz torque is given by the following equation:

$$\boldsymbol{T} = \int_{Vol} \boldsymbol{r} \times (\boldsymbol{J} \times \boldsymbol{B}) dVol$$

where:

- $\boldsymbol{r}$ is the displacement vector from the axis of rotation.
- $\boldsymbol{J}$ is the current density.
- $\boldsymbol{B}$ is the magnetic flux density.

This integral is computed for the x-, y-, and z-axes, giving the net torque on the object(s) about each axis of rotation.

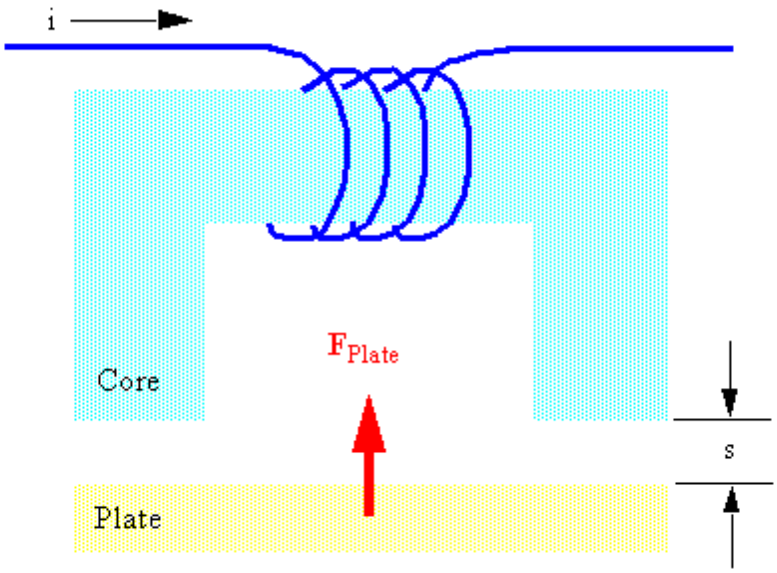
Note	The Lorentz torque does not compute the correct torque on objects that are assigned materials with a relative permeability greater than one (that is, materials where $\mu \neq \mu_0$). To find the torque on these objects, use virtual torque.

Virtual Force

To compute the virtual force on an object, the system uses the principle of virtual work. In the structure shown below, the force on the plate in the direction of the displacement, s, is given by the following relationship:

$$\boldsymbol{F}_{plate} = \left. \frac{dW(s,i)}{ds} \right|_{i = Constant}$$

where W(s,i) is the magnetic coenergy of the system. The current, i, is held constant.



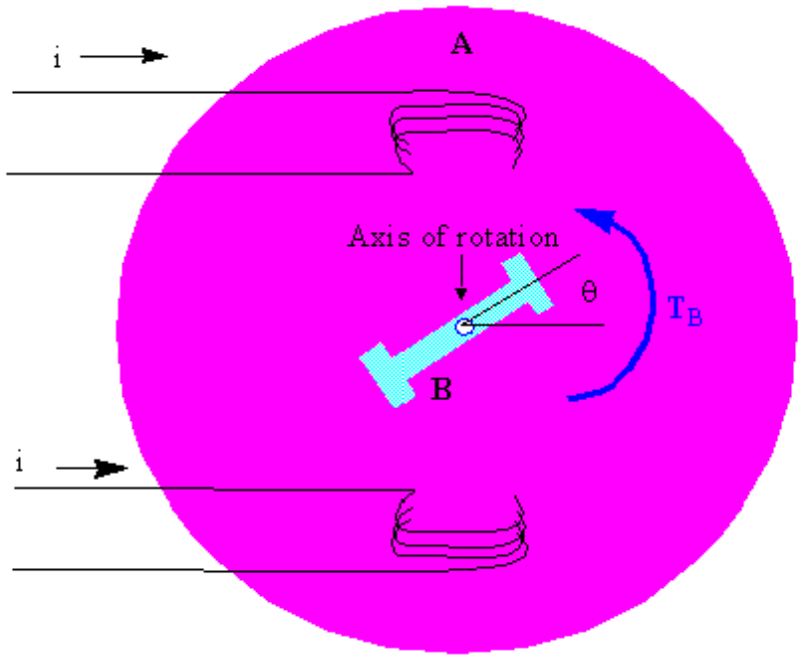
Unlike the classical virtual work method, the plate is not actually moved during the force computation. Instead, only the tetrahedra that lie along the outside surface of the object are virtually distorted. Thus, the force computation only requires one field solution.

Virtual Torque

Similar to the virtual force calculation, the system uses virtual work principles to compute the torque on an object. In the structure shown below, the torque on Object B about the axis of rotation is given by the following relationship:

$$T_B = \left. \frac{dW(\theta, i)}{d\theta} \right|_{i = Constant}$$

where W(θ, i) is the magnetic coenergy of the system. The current, i, is held constant.



Unlike the classical virtual work method, Object B is not actually rotated during the force computation. Instead, only the tetrahedra that lie along the outside surface of the object are virtually distorted. Thus, the change in the system's coenergy (and therefore the virtual torque) is given by the change in the coenergy of these tetrahedra.

Frequency Domain (Eddy Current) Solver

$\vec{T}-\Omega$

The **3D eddy current solver (frequency domain or harmonic solver)** uses the formulation. It is based on the assumption that all electromagnetic fields pulsate with the same frequency (specified by the user) and have magnitudes and initial phase angles calculated by Maxwell. There are no moving objects (velocity is zero everywhere). Permanent magnets cannot be part of the model, and all materials are assumed to be linear theoretically. For designs with nonlinear materials, analysis is based on the fundamental components of B and H at a specified frequency as a proximation. See Nonlinear eddy field simulation for more information.

Electromagnetic radiation can also be simulated.

In the non-conducting regions the magnetic field strength is given by the following equation:

$$\vec{H} = \vec{H}_p + \nabla\Omega$$

using node element with the same quadratic approximation as for the case of the magnetostatic solver.

In regions with non-zero conductivity where eddy current calculation has been set up, the following equation is true:

$$\vec{H} = \vec{H}_p + \nabla\Omega + \vec{T}$$

where

$\vec{T}$

is the electric vector potential calculated using edge elements. At the interface between conductors and non-conductors the tangential component of the electric vector potential is constrained to zero.

The quantity calculated by Maxwell is in this case the magnetic field H(x,y,z).

For eddy current problems, typical sources associated with non-winding excitation are Current and Current Density; typical sources associated with winding excitation are Current and Voltage. In applying the sources for the magnetic field problems, keep in mind that the applied current distribution must be divergence free in the entire space of the solution as it is physical for (quasi) stationary conduction current density distributions. Thus, the conduction paths(s) for the applied current distributions must be closed when totally contained within the solution space for the problem or must begin and end at the boundaries. The total current applied to conductors that touch the boundaries doesn't require the existence of terminals at the ends where the current is applied, the respective planar surfaces of the conductors in the plane of the region (background) can be used to apply the excitations.

Typical boundary conditions used in eddy current problems include **magnetic field tangent** (the default, natural boundary condition which is automatically applied on all surfaces of the problem space -the surfaces of the geometry entity containing the model inside it). This default boundary condition can be overwritten if other boundary conditions are applied on exterior surfaces of the solution space. The default boundary condition confines the magnetic field to the solution space and thus this boundary must be placed at some distance from the sources of the problem to avoid over-constraining the fields by placing the boundaries to close to the model objects. While it is difficult to provide "recipes" with general validity on the placement of the boundaries of problems, a good rule of thumb says that if a model can be imagined as being contained in a sphere of radius R, the boundaries can be placed at a 4-5 radii R from the imaginary center of the model.

The **Zero Tangential H Field** allows you to prescribe a normal (in average) magnetic field orientation on an arbitrary surface. This boundary condition does not require any further user input, that is, no value and coordinate system specification are necessary.

In the case of **Tangential H Field** boundary condition the values of two tangential complex components (real and imaginary) and a surface coordinate system have to be defined and subsequently used in the assignment process of this boundary condition. This boundary condition

is restricted to two kinds of surfaces: planar or cylindrical. The CS in the planar case would be a rectangular coordinate system with X and Y components of H in the plane. The coordinate system in the cylindrical case would be a coordinate system with a Z axis coinciding with the axis of the cylinder and the two components to input would be the PHI and Z axis components of H.

In the case of the tangential H field caution is advised: this boundary condition should be specified such that Ampere's theorem is not violated inside the field domain or at the boundaries. This boundary condition is very useful to simulate the behavior of devices "immersed" in an electromagnetic filed of desired magnitude and orientation.

Symmetry boundary conditions are used to solve problems with symmetry and thus allows the users to take advantage of the significant reduction of the problem size for a given accuracy. Symmetry boundary conditions are of two kinds, **Odd (flux tangent)** or **Even (Flux normal)**. Although the use of this symmetry boundary condition overlaps with previous conditions to some extent, the symmetry boundary conditions should only be used in clear symmetry cases.

The **Insulating** boundary condition is of particular use for applications where very thin insulating layers are impractical to model due for example to high aspect ratio geometries that would be generated and associated meshing difficulties. Thus insulating boundary condition can be assigned to surfaces of separation between conductors. Another remarkable situation where such a boundary condition proves to be extremely useful is in modeling faults in conductors (cracks for example). Thus, modeling 2D (sheet) objects at the location of the respective cracks and applying the insulation boundary condition proves to be a very effective way of modeling the flaws without having to generate 3D cracks that would be difficult and impractical to mesh.

Phasor Notation for an Eddy Current Solution

Time varying quantities that have the form:

$$\boldsymbol{F}(t) = F_m \cos(\omega t + \theta)$$

can be represented as rotating phasors in the complex plane. Using Euler's formula:

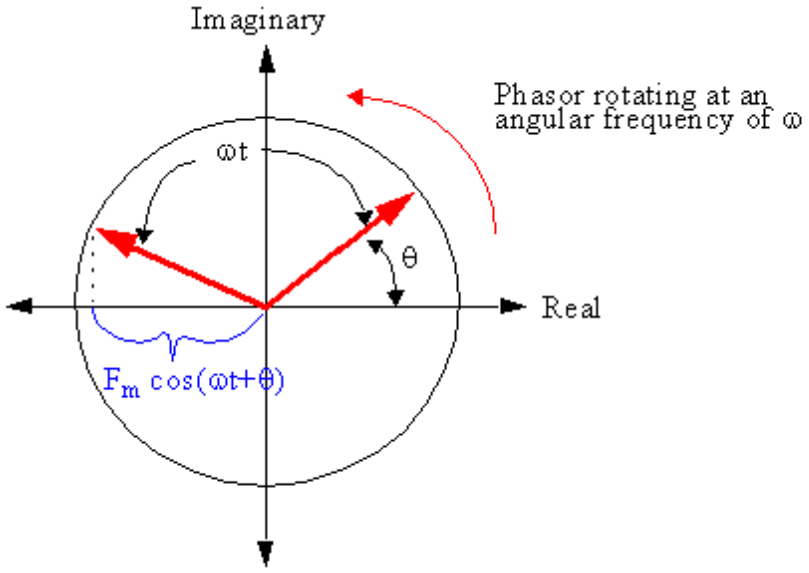
$$e^{j\alpha} = \cos\alpha + j\sin\alpha$$

$$e^{j(\omega t + \theta)}$$

If a = w*t*+q, **F**(t) equals the real portion of :

$$F(t) = \Re\left[F_m e^{j(\omega t+\theta)}\right] = \Re[F_m(\cos(\omega t+\theta)+j\sin(\omega t+\theta))] = F_m\cos(\omega t+\theta)$$

Each time-varying quantity has the form $F_m e^{j\theta} e^{j\omega t}$. The component $F_m e^{j\theta}$ is a complex constant that can be represented by a stationary phasor in the complex plane. The component $e^{j\omega t}$ is a complex number that depends on *t* and can be represented as a rotating phasor in the complex plane. The phasor's projection on the real axis oscillates sinusoidally. It reaches a peak when parallel with the real axis, and crosses zero when parallel with the imaginary axis. Therefore, a phasor with $\theta=90^o$ represents a quantity that peaks 90 degrees after a phasor with $\theta=0^o$.

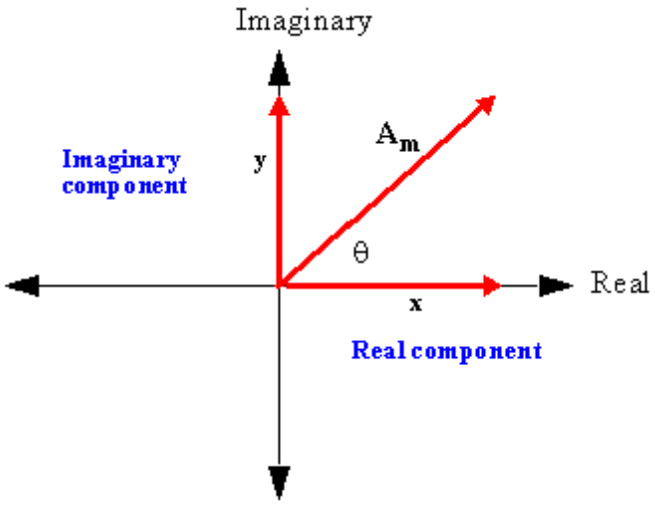


Real and Imaginary Components for an Eddy Current Solution

In general, you must enter a magnitude and phase for all AC voltages, currents, and other boundary or source quantities. But if the magnitude and phase angle of a quantity are functions of position, specify the functions in terms of real and imaginary components (x+jy). The "x+jy" description of a phasor indicates that the phasor is the sum of two components - a sinusoid that peaks at $\omega t=0°$and a sinusoid that peaks at $\omega t=90°$.

- The "x" component of the phasor is the real component and is represented by a phasor that lies on the real axis of the complex plane.
- The "y" component is the imaginary component and is represented by a phasor that lies on the j-axis of the complex plane.

The real and imaginary components of a three phase system are shown below:



They are related to the magnitude and phase of a sinusoid in this way:

$$A_m e^{j\theta} = x + jy = A_m(\cos\theta + j\sin\theta)$$

$$A_m = \sqrt{x^2 + y^2}$$

$$\theta = \text{atan}\left[\frac{y}{x}\right]$$

where the real and imaginary components are:

$$Re = A_m \cos\theta$$

$$Im = A_m \sin\theta$$

For a symmetric three-phase system, θ = 0, 120, 240 degrees, respectively.

For a frequency domain simulation (eddy current), all sources (currents, fields) must have the same frequency. Therefore, the simulation frequency is specified as a global input parameter.

In simulations where the electromagnetic radiation is included the displacement, currents are included in the calculation. In such a situation, the dependency between H and J is given by the following equation:

$$\nabla \times \vec{H} = J + \frac{\partial}{\partial t}(\varepsilon \vec{E})$$

Solution Process for an Eddy Current Solution

To solve for the magnetic field, **H**, the solver computes the values as follows:

$$\nabla \times \left(\frac{1}{\sigma + j \cdot \omega \cdot \epsilon} \nabla \times H\right) = -j \cdot \omega \cdot \mu \cdot H$$

In conductors where eddy currents occur, **H** is computed directly from source currents and applied magnetic fields.

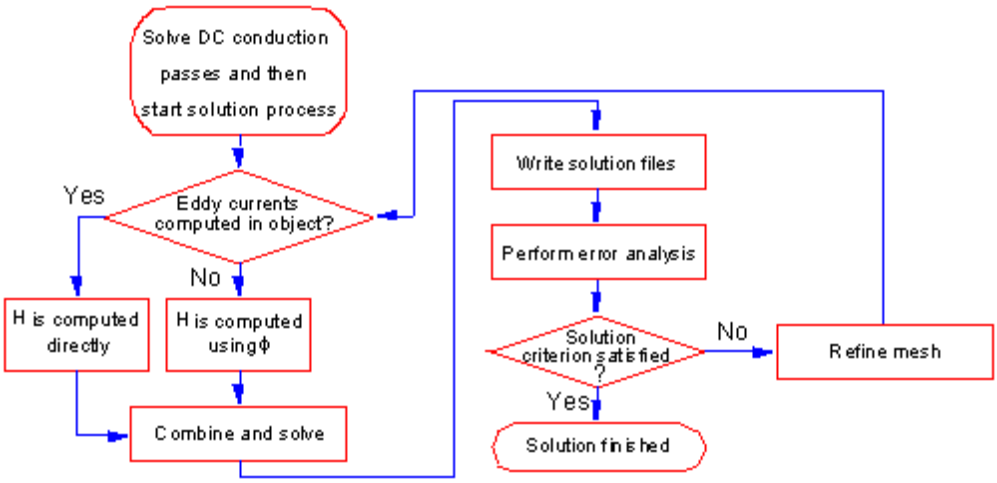
In non-conducting regions, H is computed from the magnetic scalar potential:

$$\nabla \bullet (\mu \nabla \phi) = 0$$

Directly solving for **H** requires more computing resources than using the magnetic potential, so this is done only in regions where the magnetic potential cannot be used.

The solver combines the solutions and solves for the magnetic field. **H** is forced to be continuous on the boundaries, producing a continuous field solution throughout the model. It then saves the completed solution to a file and performs an error analysis. In an adaptive analysis, it refines the tetrahedra with the highest error and continues solving until the stopping criterion is met.

Note	Always use the peak value in the applied excitations in an eddy current solution!



Nonlinear Eddy Current Field Simulation (3D)

Refer to the following topics for detailed technical information on nonlinear 3D Eddy current field simulation .

"Nonlinear Eddy Current Field Simulation" on page 28-34

"Nonlinear Eddy Current Theory " on page 28-35

"Sinusoidal B " on page 28-35

Skin Depth for an Eddy Current Solution

Induced currents allow magnetic fields to penetrate conductors only to a certain depth, of about 4-5 skin depths. The skin depth is approximated by the following formula:

$$\delta = \sqrt{\frac{2}{\omega \sigma \mu_0 \mu_r}}$$

where:

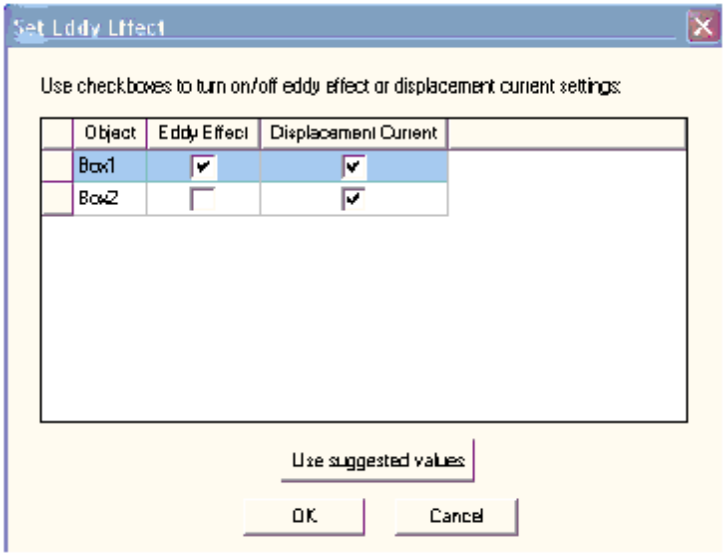
- ω is the angular frequency, which is equal to $2\pi f$ (f is the frequency at which source currents and voltages oscillate during the solution).
- σ is the conductor's conductivity in siemens/meter.
- μ_r is the conductor's relative permeability.
- μ_0 is the permeability of free space, which is equal to 4 x 10-7 H/m.

Currents are concentrated near the surface of the conductor, decaying rapidly past the skin depth. As the formula above indicates, the skin depth gets smaller as the frequency increases.

Note	Due to the skin concentration of current, AC inductances and resistances are not equal

	to their DC equivalents. This affects the inductance and resistance values computed during impedance computations.

Both the skin depth / proximity effects and the displacement effects must be specifically requested. Under the excitation setup, the respective effects can be specified for objects in the model as shown below:



Eddy (skin/proximity) effect can only be specified for objects with non-zero conductivity.

By default, eddy/displacement current effects are turned off. Including these in the calculation has a potentially significant impact on the solution time since the electric vector potential calculation is activated, causing the size of the solved matrix to increase significantly if the regions with the electric vector potential to be calculated have a large number of finite elements.

Displacement current calculation is normally activated in dielectrics (including vacuum) and allows electromagnetic waves to propagate. In such cases, a radiation boundary condition should also be used in the problem setup.

Magnetic Field Energy for an Eddy Current Solution

The energy density in the general case includes the magnetic and electric energy densities. The AC magnetic field energy is given by the following:

$$U = \frac{1}{4} Re(\boldsymbol{B} \bullet \boldsymbol{H}^* + (\boldsymbol{E} \bullet \boldsymbol{D}^*))$$

where:

- $\boldsymbol{B}$ is the magnetic flux density.
- $\boldsymbol{H}$ is the magnetic field.

- ***E*** is the electric field.
- ***D*** is the electric displacement.
- *Re* is the real part operation.
- The superscript * denotes complex conjugate.
- The symbol · is the dot product.

For the sake of brevity, the energy density in the post processor is labeled as **Energy**. The total energy (U) of an object or the whole arrangement can be calculated by integrating the energy density over a proper volume in the Field Calculator, as follows:

$$U = \frac{1}{4}\iiint_{Vol} Re(\boldsymbol{B} \bullet \boldsymbol{H}^* + (\boldsymbol{E} \bullet \boldsymbol{D}^*))dVol$$

This represents the average energy over time, not the instantaneous energy at a specific point in the cycle. (The 1/4 factor is explained by the use of peak values for excitations.)

Hysteresis Loss for an Eddy Current Solution

The hysteresis loss is given by: the following

$$U = -\int_{Vol} \omega Im\left[\frac{1}{2}(\boldsymbol{B} \bullet \boldsymbol{H}^*)\right]dVol$$

where:

- ***B*** is the magnetic flux density.
- ***H**** is the complex conjugate of the magnetic field, ***H***.
- ω is the angular frequency, equal to $2\pi f$ (where *f* is the solution frequency).

The magnetic hysteresis loss density (p_h) is given by the following:

$$p_h = \frac{1}{2}\omega Im(B \cdot \boldsymbol{H}^*)$$

where:

- *Im* is the imaginary part.
- ω is the angular frequency.

For the sake of brevity, the hysteresis loss density in the post processor is labeled as **Hysteresis Loss**. The total magnetic hysteresis loss (p_h) of an object or the whole arrangement can be

calculated by integrating the hysteresis loss density over a proper volume in the Field Calculator, as follows:

$$p_h = -\frac{1}{2}\omega \iiint_{Vol} Im(B \cdot \boldsymbol{H}^*) dVol$$

Hysteresis loss applies if the structure includes materials with a non-zero value for the tangent of the magnetic loss angle (equivalent to a non-zero imaginary permeability). Otherwise, it is equal to zero.

Ohmic Loss for an Eddy Current Solution

Ohmic loss is given by:

$$P = \int_{Vol} \frac{\boldsymbol{J} \bullet \boldsymbol{J}^*}{2\sigma} dVol \qquad \text{(Watts)}$$

where:

- ***J*** is the current density.
- ***J**** is the complex conjugate of the current density.
- σ is the conductivity in siemens/meter.

Ohmic loss is used to compute the power loss in a structure ($P=I^2R$).

For impedance boundaries, ohmic loss is given by:

$$P = \sqrt{\frac{\omega \mu_0 \mu_r}{8\sigma}} \int_{Sur} \boldsymbol{H}_t \bullet \boldsymbol{H}_t^* \, ds \text{ (Watts)}$$

where:

- *w* is the angular frequency, which is equal to 2πf. (f is the frequency at which source currents and voltages oscillate during the solution.
- *s* is the conductor's conductivity in siemens/meter.
- μ_r is the conductor's relative permeability.
- μ_0 is the permeability of free space, which is equal to 4π x 10–7 H/m.

- H_t is the tangential component of H on the impedance boundary.
- H_t^* is the complex conjugate tangential component of H on the impedance boundary.

The ohmic loss density (p_o) is given by the following:

$$p_o = \frac{1}{2} Re(\mathbf{E} \cdot \mathbf{J_c}^*)$$

This loss is calculated from the conduction component of the current density:

$$\mathbf{J}_c = [\sigma]\mathbf{E}$$

For the sake of brevity, the ohmic loss density in the post processor is labeled as **Ohmic Loss**. The total ohmic loss (p_o) of an object or the whole arrangement can be calculated by integrating the ohmic loss density over a proper volume in the Field Calculator, as follows:

$$p_o = \frac{1}{2} \iiint_{Vol} Re(\mathbf{E} \cdot \mathbf{J_c}^*) d\,Vol$$

Dielectric Loss for an Eddy Current Solution

The dielectric loss density (p_d) is as follows:

$$p_d = \frac{1}{2} Re[\mathbf{E} \cdot (\mathbf{J} - \mathbf{J_c})^*]$$

The current density and the conduction current density are defined by the following two equations:

$$\mathbf{J} = \nabla \times \mathbf{H}$$

$$\mathbf{J}_c = [\sigma]\mathbf{E}$$

For the sake of brevity, the dielectric loss density in the dielectric loss density in the post processor should be labeled as **Dielectric Loss**. The total dielectric loss (p_d) of an object or the whole arrangement can be calculated by integrating over a proper volume in the Field Calculator, as follows:

$$p_d = \frac{1}{2}\iiint_{Vol} Re[\mathbf{E} \cdot (\mathbf{J} - \mathbf{J_c})^*]\, dVol$$

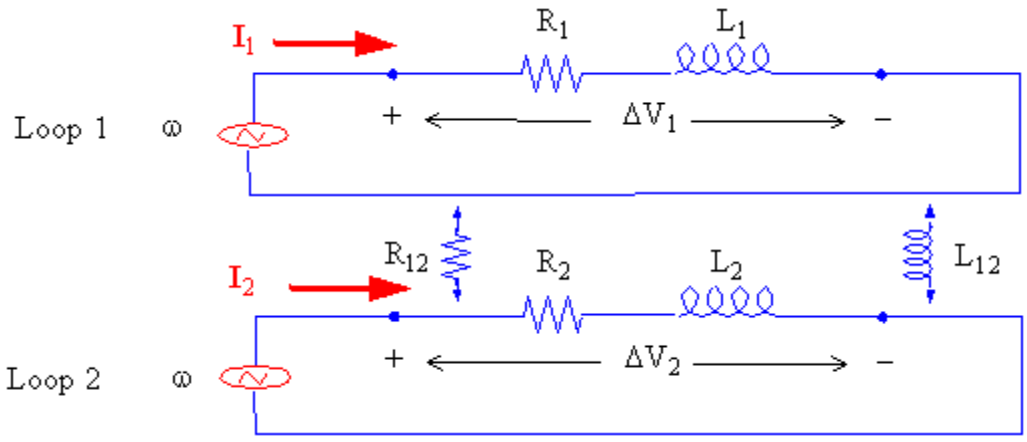
The dielectric loss can also be called *electric hysteresis loss*.

Impedance Matrix for an Eddy Current Solution

An impedance matrix gives the relationship between AC voltages and AC currents for multiple conductors. In the current loops below, the voltage and current in each loop is:

$$\Delta V_1 = I_1 R_1 + I_1 j\omega L_{11} + I_2 j\omega L_{12}$$

$$\Delta V_2 = I_2 R_2 + I_2 j\omega L_{22} + I_1 j\omega L_{12}$$



This can be expressed in matrix form as:

$$\begin{bmatrix} \Delta V_1 \\ \Delta V_2 \end{bmatrix} = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{12} & Z_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix}$$

where:

- ΔV_i and I_i are phasors.
- $Z_{11} = R_1 + j\omega L_{11}$ (the self-impedance of Loop 1).
- $Z_{12} = R_{12} + j\omega L_{12}$ (the mutual impedance between Loops 1 and 2).
- $Z_{22} = R_2 + j\omega L_{22}$ (the self-impedance of Loop 2).

A device with n current loops would have an $n \times n$ impedance matrix.

Matrix Elements for an Impedance Matrix

All impedances are complex numbers in the form:

$$Z = R + j\omega L$$

where:

- ω is the angular frequency of the AC voltages and currents, which is equal to $2\pi f$ (where f is the solution frequency).

- *R* is the resistance, given in ohms.
- *L* is the inductance, given in henries.

If one amp flows in Loop 1 and zero amps in Loop 2, the impedance matrix becomes:

$$\begin{bmatrix} \Delta V_1 \\ \Delta V_2 \end{bmatrix} = \begin{bmatrix} Z \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} Z_{11} \\ Z_{12} \end{bmatrix}$$

Diagonal Elements for an Impedance Matrix

The self-impedance of Loop 1 is given by $Z_{11} = R_1 + j\omega L_{11}$.

- R_1 represents the internal resistance of the current loop.
- L_{11} represents the self-inductance of the loop. The $j\omega L_{11}$ term represents the inductive reactance of the loop — the apparent AC inductance due to the loop's self inductance.

Off-Diagonal Elements for an Impedance Matrix

The mutual impedance between Loops 1 and 2 is given by $Z_{12} = R_{12} + j\omega L_{12}$.

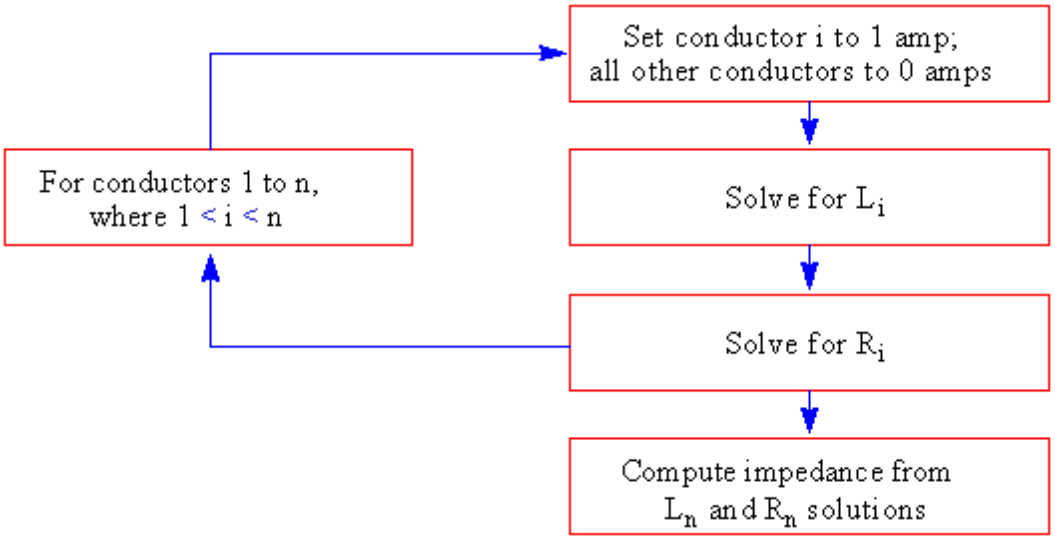
- R_{12} represents the mutual resistance of a neighboring conductor as seen from the source conductor.
- L_{12} represents the mutual inductance between the loops. The $j\omega L_{12}$ term represents the inductive reactance between the loops — the apparent AC inductance due to the mutual inductance.

Symmetry for an Impedance Matrix

The impedance matrix is symmetric about the diagonal. This indicates that the mutual effects between any two loops are identical.

Solution Process for an Impedance Matrix

The simulator divides the impedance matrix computation into two parts, as shown below.



The solver generates a field solution for each conductor in the matrix. Each conductor is assumed to be part of an independent current loop. In the first solution, the current in the first conductor is set to one amp; currents in the other conductors are set to zero. This is done by imposing current sources on the conductors. In the second solution, the current in the second conductor is set to one amp and all other conductors are set to zero amps, and so forth. Objects that are not included in the impedance matrix are not affected.

Using the field solution as input, the simulator performs inductance and resistance computations. When it finishes solving for the inductance and resistance, the simulator combines them to form the impedance matrix, using the relationship:

$$Z_i = R_i + j\omega L_i$$

Inductance for an Impedance Matrix

The inductance solution for an impedance computation is similar to a magnetostatic inductance solution. The system computes the following for each matrix entry:

$$L_{ij} = \int \boldsymbol{B}_i \bullet \boldsymbol{H}_j^* d\Omega$$

where:

- ***B*** is the magnetic flux density.
- ***H**** is the complex conjugate of the magnetic field.

Resistance for an Impedance Matrix

To find the resistance, the system computes the ohmic loss, P, for each loop:

$$P = \frac{1}{2\sigma}\int \boldsymbol{J} \bullet \boldsymbol{J}^* \, d\Omega$$

where **J** is the current density. Ohmic loss can also be expressed in terms of resistance and total current, $P=RI_{RMS}^2$, where:

$$I_{RMS} = \frac{I_{Peak}}{\sqrt{2}}$$

The resistance, therefore, is:

$$R = \frac{P}{I_{RMS}^2} = \frac{\frac{1}{2\sigma}\int \boldsymbol{J} \bullet \boldsymbol{J}^* \, d\Omega}{I_{RMS}^2} = \frac{\int \boldsymbol{J} \bullet \boldsymbol{J}^* \, d\Omega}{\sigma I_{Peak}^2}$$

To solve this, the system computes the conduction current, **J**, for each conductor. Each subsolution, $\mathbf{J}_i$, represents the contribution of a current loop toward the matrix. After it solves for **J** for all conductors, the system computes the resistance matrix.

Note	Since all current loops are assumed to be independent, an impedance matrix only contains terms for their internal resistance (or self-resistance). It does not include terms for the mutual resistances between loops.

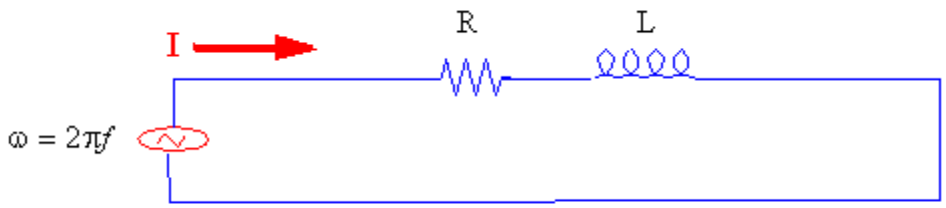
Line Impedance

Line impedance is represented by a 1x1 impedance matrix. That is to say, the line impedance is the impedance of a single current loop given by:

$$Z = R + j\omega L$$

where:

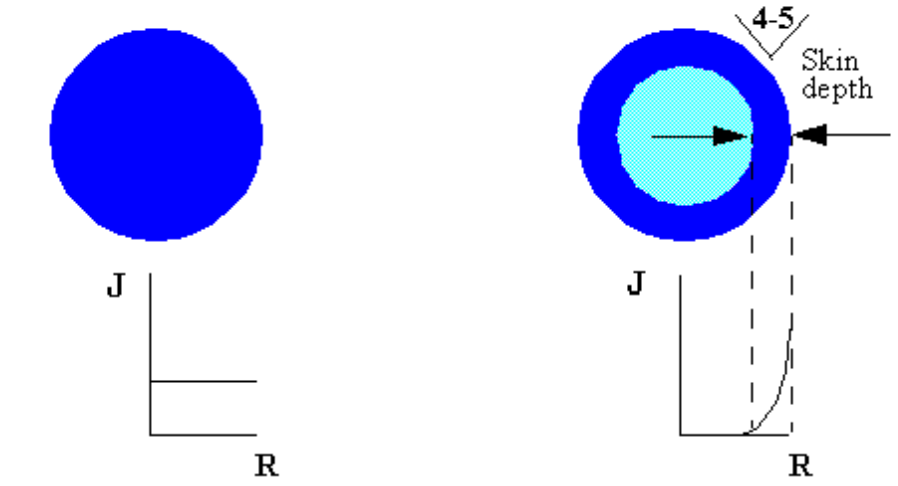
- ω is the angular frequency of the AC voltages and currents, which is equal to $2\pi f$ (where f is the solution frequency).
- R is the self-resistance of the loop.
- L is the self-inductance of the loop.



The inductance and resistance components of line impedance are computed separately.

AC Inductance and Resistance for an Impedance Matrix

The inductances and resistances computed during an impedance solution are different from those computed for the equivalent DC case, as shown below. The darker blue area represents the cross-section of the current flowing in the conductors.



DC: **J** is uniformly distributed throughout the conductor.

AC: **J** is concentrated near the surface and decays exponentially toward the conductor's center.

The current density, **J**, in the DC case (the conductor on the left) is evenly distributed throughout the cross-section of the conductor. The current density in the AC case (the conductor on the right) is distributed close to the surface due to skin concentration of currents. Since the area through

which current can flow is smaller, it follows that the resistance to the current flow is higher in the impedance matrix than in the equivalent DC case.

In the DC example, no eddy currents occur. The magnetic field created by the current flowing through the conductor is static. In the AC example, the oscillating magnetic field induces currents in other conductors in the model. These induced currents affect the computation of inductance for the impedance matrix, causing it to be different from the equivalent DC inductance.

Lorentz Force in an Eddy Current Solution

The Lorentz force represents the average force on an object (or group of objects) due to currents in a time-varying magnetic field. Lorentz force is given by:

$$\boldsymbol{F}(t) = \int_{Vol} \boldsymbol{J}(t) \times \boldsymbol{B}(t)\, dVol$$

where:

- ***J(t)*** is the current density.
- ***B(t)*** is the magnetic flux density.

The average force is found by integrating the instantaneous force:

$$\boldsymbol{F}_{AV} = \frac{1}{2\pi}\int_{0}^{2\pi} \boldsymbol{F}_{Inst}\, d\omega t = \frac{1}{2\pi}\int_{0}^{2\pi}\left(\int_{Vol} \boldsymbol{J}(t) \times \boldsymbol{B}(t)\, dVol\right) d\omega t$$

Warning	The Lorentz force does not compute the correct average force on objects that are assigned materials with a relative permeability greater than one. To find the force on these objects, use virtual force.

Lorentz Torque in an Eddy Current Solution

The Lorentz torque represents the average torque on an object or group of objects due to currents in a time-varying magnetic field. The Lorentz torque is given by:

$$\boldsymbol{T}(t) = \int_{Vol} \boldsymbol{r} \times [\boldsymbol{J}(t) \times \boldsymbol{B}(t)]\, dVol$$

where:

- ***r*** is the displacement vector from the axis of rotation.
- ***J(t)*** is the current density.
- ***B(t)*** is the magnetic flux density.

The average torque is found by integrating the instantaneous torque:

$$\boldsymbol{T}_{AV} = \frac{1}{2\pi}\int_0^{2\pi} \boldsymbol{T}_{Inst}\, d\omega t = \frac{1}{2\pi}\int_0^{2\pi}\left(\int_{Vol} \boldsymbol{r} \times [\boldsymbol{J}(t) \times \boldsymbol{B}(t)]\, dVol\right) d\omega t$$

This integral is computed for the x-, y- and z-axes, giving the average torque on the object(s) about each axis of rotation.

Warning	The Lorentz torque does not compute the correct average torque on objects that are assigned materials with a relative permeability greater than one. To find the torque on these objects, use virtual torque.

Virtual Force in an Eddy Current Solution

Virtual force in an eddy current problem is computed the same way as the magnetostatic virtual force. The only difference is that the average force over time is computed — not the net (instantaneous) force at a given time. The average virtual force is found by integrating the instantaneous force:

$$\boldsymbol{F}_{AV} = \frac{1}{2\pi}\int_0^{2\pi} \boldsymbol{F}_{Inst}\, d\omega t = \frac{1}{2\pi}\int_0^{2\pi}\left(\frac{dW(s,i)}{ds}\right) d\omega t$$

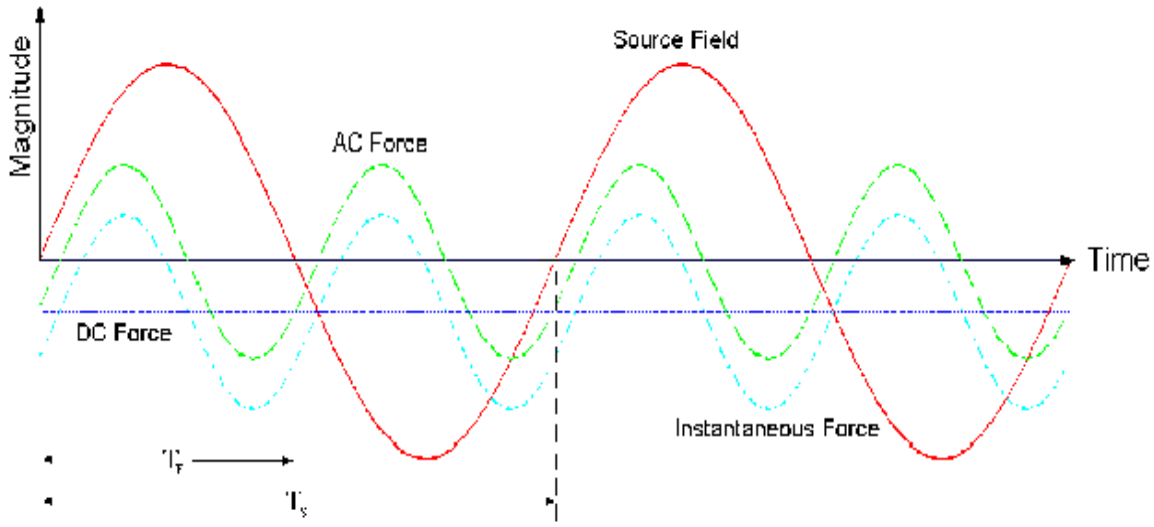
Virtual Torque in an Eddy Current Solution

Virtual torque in an eddy current problem is computed the same way as the magnetostatic virtual torque. The only difference is that the average torque over time is computed — not the net torque at a given time. The average torque is found by integrating the instantaneous torque:

$$\boldsymbol{T}_{AV} = \frac{1}{2\pi}\int_0^{2\pi} \boldsymbol{T}_{Inst}\, d\omega t = \frac{1}{2\pi}\int_0^{2\pi}\left(\frac{dW(\theta,i)}{d\theta}\right) d\omega t$$

Average Force in an Eddy Current Solution

The Lorentz force and virtual force computed for an eddy current model represent the average force, not the instantaneous force at a specific point in the AC magnetic field's cycle. Similarly, the Lorentz torque and virtual torque represent the average torque over time, since these torque computations make use of the time-averaged force. The difference between the time-averaged (or DC) force, AC force, and instantaneous force is shown below:



Force oscillates at twice the frequency of the source current and magnetic field:

$$f_F = \frac{1}{T_F} = 2f_S$$

where:

- f_F is the frequency of the force.
- f_S is the frequency of the source current and magnetic field.
- T_F is the period of the force.

The time-averaged (or DC) force, AC force, and instantaneous force can be determined by:

$$F_{DC} = \frac{1}{2}\int Re\left|\bar{J} \times \bar{B}^{*}\right| dV$$

$$F_{AC} = \frac{1}{2}\int Re\left|\bar{J} \times \bar{B}\right| dV$$

$$F_{INST} = F_{DC} + F_{AC}$$

The AC force, F_{AC} must be evaluated at a particular phase (=ωt) in order to determine its magnitude at an instant in time.

3D AC Conduction Solver

AC conduction analysis is the study of:

- electric field
- losses arising in conductors and imperfect (lossy) dielectrics from the application of an alternating (AC) voltage or external current to the electrodes.
- C and G parameters extraction

In this harmonic formulation, the magnetic fields are considered negligible, and the electromagnetic field can be treated as electro-quasistatic. The electric field is derived from the electric scalar potential φ.

One can use this analysis to determine the voltage, electric field, electric flux density, and electric current density distributions, capacitive effects, extract the C and G parameters in electric devices as a function of frequency in response to time-harmonic loading.

Complex arithmetic is used to represent harmonic time-dependency as outlined below.

Governing equations

Given the Ampere's Law in frequency domain,

$$\nabla \times \boldsymbol{H} = \boldsymbol{J_c} + j\omega \boldsymbol{D}$$

it is possible to obtain the continuity equation solved by the 3D AC conduction solver,

$$\nabla \cdot \left(\mathbf{J_c} + j\omega \mathbf{D}\right) = 0$$

where:

H is the magnetic field

$\mathbf{J_c}$ is the conduction current density

ω is the angular frequency at which the potential is oscillating

D is the displacement flux density

Constitutive relations

The constitutive relations are given by:

$$\mathbf{J}_c = \sigma_s \, (E) \; \mathbf{E}$$

and

$$\mathbf{D} = \dot{\varepsilon} \;\; (E) \; \mathbf{E}$$

where:

σ_s is the static field electrical conductivity

$\dot{\varepsilon}$ is the complex dielectric permittivity: $\dot{\varepsilon} = \varepsilon' - j\varepsilon''$

Continuity equation including lossy domains

Applying the above constitutive relations in the continuity equation, one obtains:

$$\nabla \cdot (\sigma_s \; \mathbf{E} + j\omega\dot{\varepsilon} \; \mathbf{E}) = 0$$

$$\nabla \cdot [(\sigma_s + \omega\varepsilon'') \; \mathbf{E} + j\omega\varepsilon' \; \mathbf{E}] = 0$$

$\sigma_s + \omega\varepsilon''$ represents the equivalent conductivity (σ_e) composed by a static (σ_s) and an alternating (σ_a) part, which is caused by dipole rotation when the material is submitted to an alternating electrical field.

$$\sigma_e = \sigma_s + \omega\varepsilon'' = \sigma_s + \sigma_a$$

Finally, the continuity equation can be written as:

$$\nabla \cdot [\nabla\varphi \, (\sigma_s + \omega\varepsilon'' + j\omega\varepsilon' \,)] = 0$$

It is of importance to mention that normally the dielectric lossy materials are defined in terms of the dielectric loss tangent *(tan δ)*. In this case, the alternating *tan δ* is considered:

$$\tan\delta_a = \frac{\varepsilon''}{\varepsilon'}$$

Consequently:

$$\varepsilon'' = \tan\delta_a \varepsilon'$$

Time-Harmonic Field Assumptions

The AC conduction field solver assumes the following conditions about field quantities:

- All time-varying electromagnetic quantities have the periodic waveform:

$$F(t) = F_m \cos(\omega t + \theta)$$

where all quantities must have the same value of ω, but can have different phase angles (θ).

- If a current is not a pure sinusoid, it is decomposed into sinusoidal harmonics, and solved separately at each frequency.
- The component of **E** due to time-varying magnetic fields caused by conduction currents can be neglected.

Non-linear Solution - 3D AC Conduction Solver

Unlike linear problems in which the field solution is assumed to be based on the sinusoidal complex peak field values in a steady state, nonlinear problems are usually based on all harmonic components of the fields.

The 3D AC Conduction solver supports the J-E and D-E non-linear properties, as stated in the constitutive relations:

$$\mathbf{J}_c = \sigma_s\,(E)\ \mathbf{E}$$

$$\mathbf{D} = \dot{\varepsilon}\ \ (E)\ \mathbf{E}$$

Thus for nonlinear problems, though the electric field is sinusoidal due to sinusoidal excitations, the resulting D and/or J fields can be “harmonically rich” due to the material non-linearity.

Although the D and J fields are not sinusoidal at output, the Maxwell 3D AC Conduction solver uses an effective property based on the sinusoidal E to approximate non-linear field behavior associated with the assigned non-linear JE and DE characteristic. Using the JE curve as an example, the effective conductivity is expressed by:

$$\sigma_{eff} = \frac{\frac{1}{T}\int_0^T J(|E|,t)dt}{\frac{2}{\pi}\widehat{E}}$$

The above equation is used to obtain the effective material property for each electric field value in the original curve, allowing to generate a new curve based on the effective properties. During the simulation process, the effective curve is used to calculate the conductivity and permittivity.

> **Note:** Though the complex permittivity *($\acute{\varepsilon}=\varepsilon'-j\varepsilon''$)* is considered in the constitutive relation, the D-E curve describes only its real part ε'.

Field Definitions for the 3D AC Conduction Solver

The fields can have different definitions in harmonic problems, mainly when dealing with lossy materials. In this section the fields available in the 3D AC conduction solver are presented.

Field Name	Type Definition
Voltage: electric potential φ	scalar and complex
Electric field (E)	vector and complex $\boldsymbol{E} = -\nabla\varphi$

Field Name	Type Definition
Electric flux density (D)	vector and complex $\mathbf{D} = \dot{\varepsilon}\ (E)\ \mathbf{E}$ where: $\dot{\varepsilon} = \varepsilon' - j\varepsilon''$
Total current density (J)	vector and complex $J = (\sigma_s(E) + \omega\varepsilon'' + j\omega\varepsilon'(E))E$
DC current density (Jc)	vector and complex $\mathbf{J}_c = \sigma_s\ (E)\ \mathbf{E}$
Ohmic loss	scalar and real $P_{ohmic} = \frac{1}{2} Re\left\{J_c \cdot E^*\right\}$ or $P_{ohmic} = \frac{1}{2} Re\left\{\sigma_s\ (E \cdot E^*)\right\}$
Dielectric loss	scalar and real $P_{dielectric} = \frac{1}{2} Re\left\{(J - J_c) \cdot E\right\}$ or $P_{dielectric} = \frac{1}{2} Re\left\{\omega\varepsilon''\ (E \cdot E^*)\right\}$
Energy	scalar and real $Energy = \frac{1}{4} Re\left\{\mathbf{D} \cdot \mathbf{E}^*\right\}$
Surface charge density (Qsurf)	scalar and complex $n \cdot (D_2 - D_1) = \rho_s$ or $n \cdot (\varepsilon_2 E_2 - \varepsilon_1 E_1) = \rho_s$ where • E_1 is the electric field and ε_1 the permittivity in the material on one side of the boundary. • E_2 is the electric field and ε_2 the permittivity in the material on opposite side of the boundary. • n is the normal vector of the surface.

Losses Calculation - 3D AC Conduction Solver

The average apparent power along the domain Ω is given by:

$$S = \frac{1}{2}\int_{\Omega} J^* \cdot E d\Omega$$

Consequently, the active (including Ohmic and dielectric) and reactive powers can be expressed by its real and imaginary parts, as follows:

$$P_{active} = \frac{1}{2} Re\left\{\int_{\Omega} J^* \cdot E d\Omega\right\}$$

$$P_{reactive} = \frac{1}{2} Im\left\{\int_{\Omega} J^* \cdot E d\Omega\right\}$$

Admittance - 3D AC Conduction Solver

Admittance can best be explained as the inverse of impedance, and is expressed by:

$$Y = G + j\omega C$$

where:

- ω is equal to $2\pi f$, where f is the frequency of the AC voltage source.
- Y is the admittance in Siemens.
- G is the conductance in Siemens.
- ωC is the susceptance in Siemens.

Note	The matrix values displayed by the software are conductance in ohms and capacitance in farads (not conductance and susceptance) and therefore do not include jω.

Admittance Matrix - 3D AC Conduction Solver

The electric parameters are obtained by the equivalence between the dissipated power in terms of discrete and continuous model.

In the discrete circuit, neglecting the inductive term, the average apparent power in the complex formalism is given by:

$$S = \frac{1}{2} U I^* = \frac{1}{2} V^2 Y = \frac{1}{2} V^2 \left(G + j\omega C\right)$$

In the continuous model,

$$\mathrm{S} = \frac{1}{2}\int_{\Omega} \boldsymbol{J}^* \cdot \boldsymbol{E}\, \mathrm{d}\Omega$$

Consequently, one can write:

$$S = \frac{1}{2}V^2 (G + j\omega C) = \frac{1}{2}\int_{\Omega} J^* \cdot E \, d\Omega$$

As the extraction of parameters is performed by applying a source of 1V at each port, the admittance can be obtained by integrating the power along the domain.

$$Y = G + j\omega C = \int_{\Omega} J^* \cdot E \, d\Omega$$

This process is repeated N (number of excitations) times and the parameters are obtained as follows:

$$G = Re\left\{\int_{\Omega} J^* \cdot E \, d\Omega\right\}$$

$$C = \frac{1}{\omega} Im\left\{\int_{\Omega} J^* \cdot E \, d\Omega\right\}$$

Iterative Matrix Solver Technical Details

Two iterative solvers (PCG and QMR) are supported in Maxwell. The PCG is supported in the magnetostatic and electrostatic solvers and QMR is supported in the eddy current solver.

For large simulations, the iterative solvers save significant memory - easily a factor two or more, and may be also faster than the direct solver.

Consider the matrix equation:

$Ax = b$	(1)

where A is a matrix, b a right hand side and x the solution.

When A^{-1} is computationally expensive or the exact solution x is impossible, an alternative is to seek an approximation $\tilde{x}$ to x, with an error $e = x - \tilde{x}$. The exact solution can therefore be rewritten as:

$e = x - \tilde{x}$	(2)

Substituting (2) into (1) results in the so-called residual equation:

$Ae = r$	(3)

where r is called the residual defined by:

$r = b - a\tilde{x}$	(4)

As mentioned above, the exact solution for e in (3) is impossible since it requires A^{-1}. However, if an approximation $M \approx A$ is available, the error e can be approximated in (3) by:

$\tilde{e} = M^{-1}r$	(5)

Finally, the approximation $\tilde{x}$ is updated by:

$$\tilde{x} \leftarrow \tilde{x} + \tilde{e} \quad (6)$$

It is (4)-(6) that form the foundation of the iterative solution method. A matrix solver using the iterative solution method is called an *iterative matrix solver*. The method starts with an initial guess $\tilde{x} = x_0$ and repeats (4)-(6) until the approximation $\tilde{x}$ to x is within tolerance, or the number of iterations exceeds a given number. In the former case, it is said the solution converges; while in the latter, it doesn't.

The residual r is used for measuring the closeness of $\tilde{x}$ to x. Since A and b in (1) can be scaled by the same factor without altering x, so does the residual r in (4). It typically makes more sense to replace $\|r\|$ as the stopping criterion with the relative residual:

$$res = \frac{\|r\|}{\|b\|} \quad (7)$$

where $\| \|$ stands for vector norm.

M

A
in (5) is called a preconditioner of . A good preconditioner greatly reduces the number of

M M A
iterations. The following makes a good preconditioner: is a good approximation to in

M^{-1}
some sense and is computationally cheap.

Some of the classic iterative matrix methods include:

- M A
  The Jacobi method where is the diagonal of .
- M A
  The Gauss-Seidel method where is the lower triangular or upper triangular matrix of .
- M
  The successive over-relaxation method (SOR) where is a weighted combination of the
  A
  lower triangular and upper triangular matrix of .

Refer to the following reference for the details of iterative solvers:

Henk A. van der Vorst, *"Iterative Krylov Methods for Large Linear System"* Cambridge University Press, 2003.

3D Transient Excitations (Sources)

$\vec{T}-\Omega$
In the **3D transient (time domain)**, the solver uses the formulation. Motion (translational or cylindrical/non-cylindrical rotation) is allowed, excitations - currents and/or voltages- can assume arbitrary shapes as functions of time, nonlinear BH material dependencies are also allowed. The support of voltage excitations for the windings has as consequence the fact that the winding currents are unknown and thus the formulation has to be modified slightly to allow Maxwell to account for source fields due to unknown currents in voltage - driven solid conductors (where eddy effects are evaluated) and in voltage-driven stranded conductors -where the eddy effects (such as skin and proximity effects) are ignored. Also for a simpler formulation of problems where motion is involved, Maxwell uses a particular convention and uses the fixed coordinate system for the Maxwell's equations in the moving and the stationary part of the model. Thus the motion term is completely eliminated for the translational type of motion while for the rotational type of motion a simpler formulation is obtained by using a cylindrical coordinate system with the z axis aligned with the actual rotation axis.

The formulation used by the Maxwell transient module supports Independent-Dependent boundary conditions and motion induced eddy currents everywhere in the model, in the stationary as well as in the moving parts of the model. Mechanical equations attached to the rigid-body moving parts allows a complex formulation with the electric circuits being strongly coupled with the finite element part and also coupled with the mechanical elements whenever transient mechanical effects are included by users in the solution. In this case the electromagnetic force / torque is calculated using the virtual work approach. For problems involving rotational type of motion a "sliding band" type of approach is followed and thus no re-meshing is done during the simulation. For translational type of motion the mesh in the band object (surrounding the part in motion) needs to be re-created at each time step with a degree of refinement which is dependent upon the mesh size in the moving object. In this later case the mesh in both stationary and moving objects remains unchanged as initially created by the user. For transient type of electromagnetic field analysis (with or without motion) the user is responsible for creating the mesh that is capable to "catch" the respective physics such as skin and proximity effects -if any- are to be present in the resulting fields.

The following three Maxwell's equations are relevant for transient (low frequency) applications:

$$\nabla \times H = \sigma(E)$$

$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}$$

$$\nabla \cdot B = 0$$

The following two equations directly result from the above equations:

$$\nabla \times \frac{1}{\sigma} \nabla \times H + \frac{\partial B}{\partial t} = 0$$

$$\nabla \cdot B = 0$$

The final result is a formulation where vector fields are represented by first order edge elements and scalar fields are represented by second order nodal unknowns.

Field equations are coupled with circuit equations for both solid and stranded conductors because, in the case of applied voltage supplies, the currents are unknown. For the case of voltage driven solid conductors, the following equation is used to account for the ohmic drop across the i-th conductor loop:

$$V_{Ri} = \int_{R_C(i)} \iint J_{0i}(E + v \times B)dR$$

where J_{0i} represents the current density.

The current density J0i corresponds to 1A of net current in loop i and vanishes outside loop i.

The induced voltage can be derived from the following equation:

$$e_i = -\iiint_{R_C(i)} H_i \bullet B dR$$

where the integration is performed over the whole conductor region.

Stranded conductors are considered to be without induced eddy currents and, thus, are placed in the non-conducting region. This means that, for the purpose of calculating the ohmic voltage drop, we cannot use the same procedure we use for solid conductors. Instead, we use a lumped parameter to represent the DC resistance of the winding. We obtain the induced voltage due to the total flux linkage in a similar way as for solid conductors. In both cases, it is also possible to add an external inductance and capacitance.

For the time discretization, a backward time stepping scheme is used:

$$\left\{\frac{\mathbf{dx}}{\mathbf{dt}}\right\}^{t+\Delta t} = \frac{\left\{x^{t+\Delta t}\right\} - \{x\}^t}{\Delta t}$$

For the nonlinearities allowed in 3D transient applications, the classical Newton-Raphson algorithm is used.

The transient solver in Maxwell supports the **coil terminals** and **winding** definitions. Thus it is possible to specify the number of turns of coils in models which is necessary for the calculation of global quantities with high engineering value such as flux linkage and back emf of coils. Thus for the 3D transient solver a number of quantities are automatically calculated and displayed as 2D plots (functions of time): voltage (current), flux linkage, back emf. Other global quantities can be

also calculated by the 3D transient solver and displayed as 2D plots such as power loss, core loss, stranded loss, electromechanical quantities such as force/torque, speed and displacement.

A few types of sources can be used in 3D transient applications. The *Coil Terminal* type is discussed here.

Related Topics

Loss Quantities

A-Phi Formulation in Maxwell 3D (Transient)

Coil Terminals

A coil terminal excitation is the preferred excitation for Maxwell transient applications. This type of excitation can be specified as functional (arbitrary functions of time) and is very flexible; 6 different combinations can be used: current, voltage, and external circuit, with either solid or stranded conductor options.

The **current, stranded conductor** option should be used when setting coils of constant cross-section made of strands of wire (with no eddy current effects to be taken into account in the coil). The current can be functional and the specific variation as a function of time can be specified by the user.

When defining the function, already defined primitives such as sin( ), cos( ), exp( ), etc. can be used. For arbitrary variations, a piecewise linear capability can also be used by defining the corresponding dataset. You should also specify the number of winding turns.

Note	When using the stranded option, coils and, more generally, each individual current path must have a constant cross-section. However, coils and conductors that may have different individual cross-sections can be connected in windings.

The **current, solid conductor** option does not carry the restriction of constant a cross-section. The current path can have varying cross-section, and can also split as long as in the end all parallel branches merge into a unique current path. You can also define a functional excitation and apply it to the winding.

The **voltage, strand** option is similar to the corresponding current setting, with the obvious difference being that now the voltage is known, while the current remains unknown. In many situations when the voltage setup is used, other terminal characteristics that influence the current calculation are also specified: for instance, series resistance, inductance, and capacitance. No eddy current effects are taken into account, so a series resistance (DC resistance) must be specified. For this type of source, the cross-section of each conductor needs to be constant, but the cross-section can differ from one coil to another, and it is possible to construct windings using coils with different cross-sections. After the solver calculates the total current, the current is uniformly distributed on the cross-section, as it always is with a stranded option.

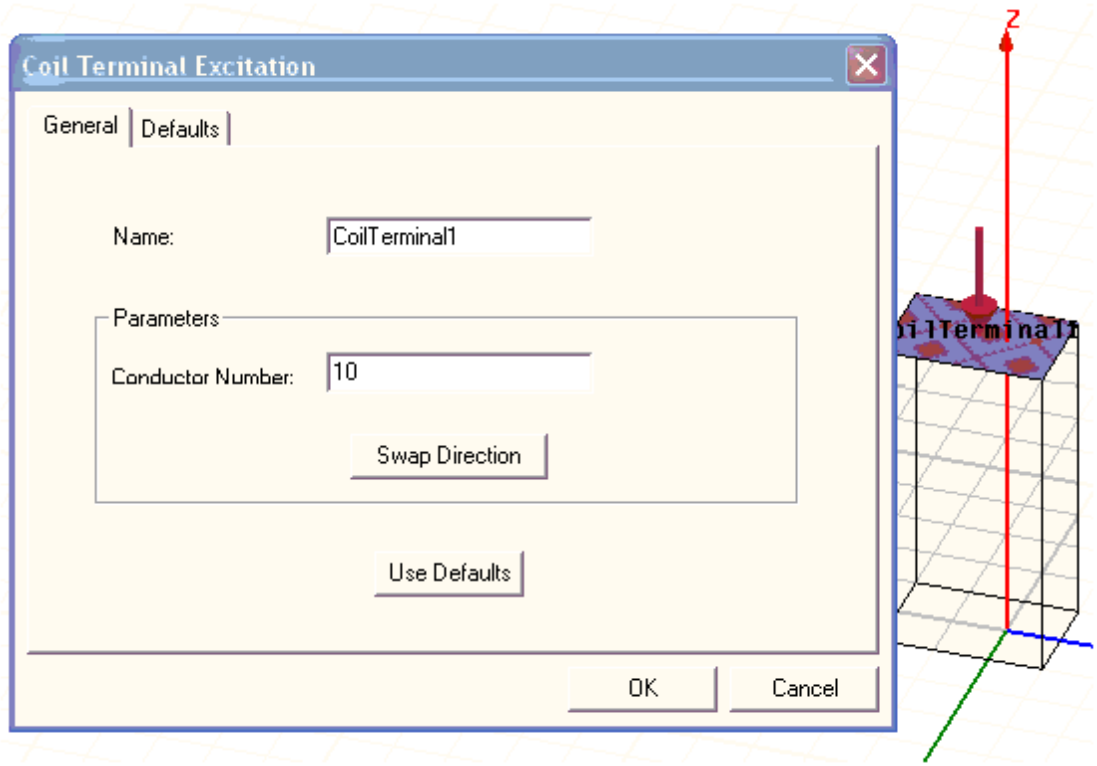
The **voltage, solid** option is chosen when solid conductors with eddy effects are part of the winding. In this situation, eddy current effects are taken into account. If required by the application,

you should also include other characteristics of the source, such as series resistance, inductance, and capacitance.

The **external strand** and **external solid** settings are used when the circuits attached to the windings have an increased degree of complexity. In the case of **external sources**, solid or stranded options are used, depending on whether or not eddy effects are to be taken into account.

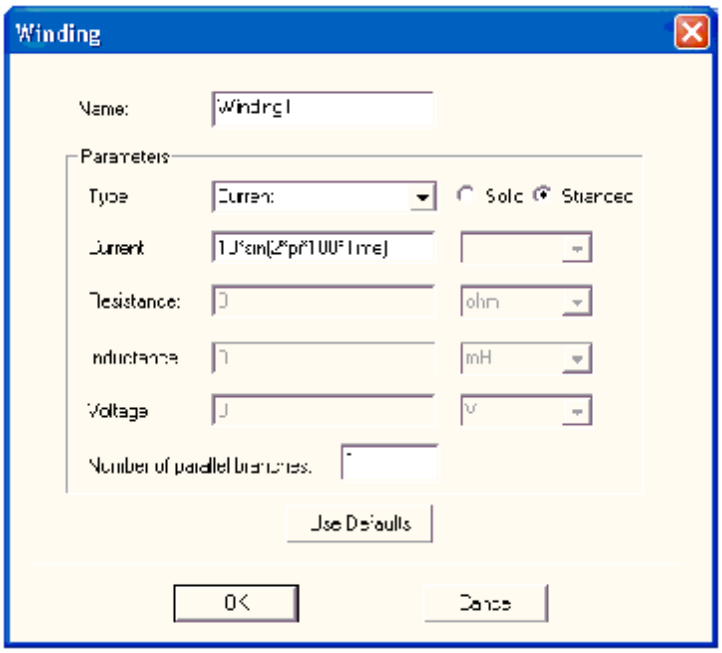
There are normally two phases in the process of defining windings.

1. First, define the terminal(s) by selecting the planar surface(s) as appropriate for the model, and then assign the coil terminal excitation by specifying the orientation of the current with respect to the terminal (current in or current out, graphically represented by an arrow). Specify the *actual* number of conductors (turns) intersecting *the chosen* planar surface (coil terminal), regardless of the symmetry (if any) of the problem.

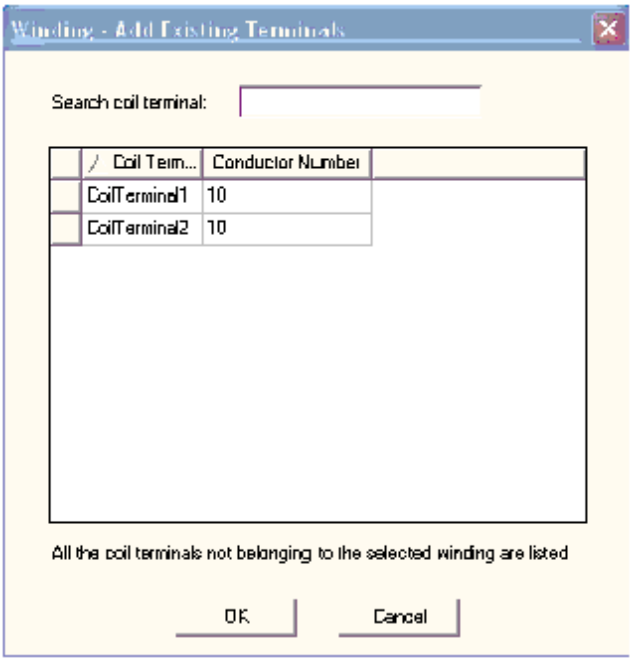


2. Right-click in the 3D Modeler window, add a new winding from the excitation menu (no need to select any geometry), and specify the remaining details of the setup (the screen shot below shows a sinusoidal current with 10 A amplitude and 100 Hz frequency being used in the setup.

Note	If external circuits are used in the excitation setup, they need to be created using the Maxwell Circuit Editor before they can be assigned to windings.



1. Right-click the winding in the property window, and specify the coil terminal(s) that belong to the winding (unassigned coil terminals are listed and can be selected from the respective window).



Solution Process for a Transient Solution

The solution process for 3D transient applications poses several challenges not apparent for other solver types. For instance, the behavior of fields is more complex than it is for static or steady-state applications, and a special finite element mesh structure is required in order for the software to accurately represent the physics.

Inductance Computation for 2D and 3D Transient Solutions

Because both 2D and 3D transient solutions are nonlinear, for inductance computation, Maxwell provides two options: *apparent inductance* and *incremental inductance*. After the FEA normal solution is completed at each time step, the permeability associated with *apparent inductance*, or the differential permeability associated with *incremental inductance*, of each element, is frozen – which is equivalent to freezing the coefficient matrix (the left side of the equation to be solved). In order to obtain winding self-inductance and mutual inductance, the solver sets an excitation current of 1A in the first winding, while setting all other winding currents to zero (permanent magnet effects are excluded). This excitation assignment corresponds to one source vector on the right-hand side of the equation to be solved. As a result, the calculated flux linkage provides the self-inductance for the first winding with 1A excitation current, and the calculated flux linkages represent mutual inductance for all other windings with zero current. Next, the solver excites the second winding with a current of 1A, while setting all other winding currents to zero (permanent magnet effects are excluded). This excitation assignment corresponds to another source vector on the right-hand side of the equation to be solved. The solver continues this process until all windings have been assigned 1A current in turn.

Related Topics

"Matrix Computation Tab Settings for Transient Solutions " on page 6-8

Flux Linkage in Maxwell 3D (Transient)

The 3D transient solver using the ***T***-Ω formulation, the turn density vector is defined as:

$$\boldsymbol{J}_k = \frac{N_k}{S_k}\boldsymbol{t}$$

where:

N_k is the number of turns;

S_k is the cross-sectional area of winding k; and

$\boldsymbol{t}$ is a unit vector tangent to the direction of winding wires.

In practice, this turn density vector is derived from the introduced basis function of the source field, $\boldsymbol{H}_k$, as:

$$\boldsymbol{J}_k = \nabla \times \boldsymbol{H}_k$$

where $\boldsymbol{H}_k$ is the field corresponding to a 1 ampere current in winding *k* and is obtained by solving a conduction problem:

$$\nabla \times \boldsymbol{H}_k = \boldsymbol{J} \qquad \nabla \times \boldsymbol{E} = 0 \qquad \boldsymbol{J} = \sigma \boldsymbol{E}$$

The turn density vector $\boldsymbol{J}_k$ corresponds to 1 ampere of net current in the winding *k*, and vanishes outside winding *k*. Thus, the induced voltage due to the total time derivative of flux linkage in winding *k* can be obtained by projecting the induced electric field *-dA/dt* onto the turn density vector $\boldsymbol{J}_k$ and integrating over the region of winding *k*:

$$e_k = -\iiint_{R_k} \boldsymbol{J}_k \bullet \frac{d\boldsymbol{A}}{dt} dR = -\iiint_{R_k} \nabla \times \boldsymbol{H}_k \bullet \frac{d\boldsymbol{A}}{dt} dR$$
$$= -\frac{d}{dt} \iiint_{R_k} \boldsymbol{H}_k \bullet \nabla \times \boldsymbol{A}\, dR - \oiint_{\Gamma} \boldsymbol{H}_k \times \boldsymbol{A} \bullet \boldsymbol{n}\, d\Gamma$$

where Γ is the boundary of a simply connected region, which is composed of the winding region R_k and the cutting region R_Σ that is introduced to fill the holes of the winding to avoid multi-valued problem of scalar potential. In such a case, since $\boldsymbol{H}_k$ has zero tangential component on boundary Γ, we have:

$$\oiint_{\Gamma} \boldsymbol{H}_k \times \boldsymbol{A} \bullet \boldsymbol{n}\, d\Gamma = 0$$

Then the induced voltage in winding *k* can be derived in terms of the $\boldsymbol{H}$ field as:

$$e_k = -\frac{d}{dt} \iiint_{R_k} \boldsymbol{H}_k \bullet \boldsymbol{B}\, dR$$

and therefore, the flux linkage in winding *k* be derived by:

$$\lambda_k = \iiint_{R_k} \boldsymbol{H}_k \bullet \boldsymbol{B}\, dR$$

Related Topics

3D Transient Excitations (Sources)

A-Phi Formulation in Maxwell 3D (Transient)

For the Maxwell 3D A-Phi transient solver, solid and stranded conductors, and flux linkage, are defined as described below.

- **Solid Conductors**

  Maxwell 3D A-Phi transient solver is formulated using the magnetic vector potential *A* and electric scalar potential φ . The relations between the potentials and the magnetic field *B* and electric field *E* are:

$$\mathrm{B} = \nabla \times \mathrm{A}$$
$$\mathrm{E} = -\frac{\mathrm{dA}}{\mathrm{dt}} - \nabla \varphi$$

  Using Generalized Ampere's law, the current continuity equation, and the constitutive relations:

$$\nabla \times \mathrm{H} = J$$
$$\nabla \cdot J = 0$$
$$J = \sigma E + \frac{\mathrm{dD}}{\mathrm{dt}}$$
$$\mathrm{B} = \mu H$$
$$\mathrm{D} = \varepsilon E,$$

  the time dependent magnetic equation is expressed as:

$$\nabla \times v \nabla \times \mathrm{A} = -\sigma \frac{\mathrm{dA}}{\mathrm{dt}} - \sigma \nabla \varphi - \frac{\mathrm{d}}{\mathrm{dt}}(\varepsilon \frac{\mathrm{dA}}{\mathrm{dt}}) - \frac{\mathrm{d}}{\mathrm{dt}}(\varepsilon \nabla \varphi) + \nabla \times H_c$$
$$\nabla \cdot (-\sigma \frac{\mathrm{dA}}{\mathrm{dt}} - \sigma \nabla \varphi - \frac{\mathrm{d}}{\mathrm{dt}}(\varepsilon \frac{\mathrm{dA}}{\mathrm{dt}}) - \frac{\mathrm{d}}{\mathrm{dt}}(\varepsilon \nabla \varphi)) = 0,$$

  where:

  H is the magnetic field intensity

  J is the total current density including source, eddy, and displacement current contributions

  σ is the conductivity

μ is the permeability

ε is the permittivity

H_c is the coercivity of the permanent magnet

$v = \mu^{-1}$ is the reluctivity.

Since Maxwell 3D is used mostly for low frequency simulations, the radiation term

$$\frac{d}{dt}\left(\varepsilon \frac{dA}{dt}\right)$$

in the above equations is ignored in the A-Phi Solver. The first order edge elements for the magnetic vector potential and the second order nodal elements for the electric potential are used. The *tree-cotree* method is used to ensure the uniqueness of the magnetic vector potential A. The Degrees of Freedom (DOFs) of edge elements on the spanning tree of the finite element mesh are set to zero. The formulation above is used in the entire problem domain except in the stranded conductors.

- **Stranded Conductors**

  For stranded conductors, the current density is assumed to be constant over the conductor domain, where the eddy and displacement effects are ignored. Therefore, only Ampere's law is considered in stranded conductors,

$$\nabla \times v \nabla \times A = J$$

  The solver approximates the constant current density distribution over each conductor for a unit current before the transient solution starts. The approximated unit current density is:

$$\hat{J} \approx \hat{t} \frac{N}{S}$$

  where:

  $\hat{t}$ is the current (winding) direction

  I is the total terminal current

  S is the area of the coil terminal into which current is entering

  N is the number of turns

  J is then written as:

$$J = \hat{J} I$$

  For winding voltage excitation, the formulation is:

$$\nabla \times v \nabla \times \mathrm{A} - \hat{\mathrm{J}}\,\mathrm{I} = 0$$
$$\iiint_{\Omega_k} \hat{\mathrm{J}} \cdot \frac{\mathrm{dA}}{\mathrm{dt}} d\Omega + (R + \mathrm{R}_{ext})I + \mathrm{L}_{ext}\frac{\mathrm{dI}}{\mathrm{dt}} = V$$

where R is the DC resistance of the winding, and R_{ext} and L_{ext} are, respectively, the external resistance and inductance.

- **Flux Linkage**

  Flux linkage calculation for windings is relatively easy compared to T-Omega formulation. It is calculated using the equation:

$$\lambda_k = \iiint_{\Omega_k} J_k \cdot A \, d\Omega$$

  where J_k is the current density in the winding under 1A excitation, A is the total magnetic vector potential in the conductor, and Ω_k is the volume of the winding.

Related Topics

3D Transient Excitations (Sources)

Complex Field Behavior for a Transient Solution

For 3D transient applications, the behavior of the fields is more complex than it is for static or steady-state applications. A diffusion of the magnetic field into the materials occurs in 3D transient situations. The distribution of the magnetic field inside objects typically has a number of spatial harmonics, which usually means the time step used in the analysis should be less (sometimes much less) than the magnetic diffusion time constant. These time constants depend upon the geometry of objects and also upon their respective material properties. For example, for a cylinder, the magnetic diffusion time constant of the fundamental spatial harmonic is given by the following equation:

$$\tau = \frac{\mu\sigma a^2}{2.4048^2}$$

where μ is the permeability of the cylinder, σ is the conductivity, and a is the radius.

Special Mesh Structure for a Transient Solution

Since eddy currents are usually considered in conductive objects, a special finite element mesh structure is required in order to accurately capture the physics. In general, a careful planning of the (manual) meshing process is required in order to achieve an accurate solution with the available hardware resources.

Overcoming Challenges for a Transient Solution

There are a number of applications where the combined effects of magnetic diffusion and strong eddy effects represent a significant challenge. Add the effect of nonlinearities that may also be present, and you have the full picture of a difficult to solve application.

The following suggestions can help you overcome these challenges presented by some 3D transient applications:

- Use symmetry whenever the problem allows it.
- Set the eddy effects only on objects where it really counts.
- In general, all unnecessary details, such as details of a geometric nature, should be eliminated, particularly in cases where a large problem is the expected task.

Boundary Conditions for a Transient Solution

The behavior of the **H**-field at the edges of the problem space is controlled through boundary conditions. The following boundary conditions can be assigned for a 3D transient model:

- Natural boundaries are assigned to the surface between objects.
- Neumann boundary conditions (homogeneous) are assigned to the outside faces of the problem region.
- Odd Symmetry (Magnetic Flux Tangential)
- Even Symmetry (Magnetic Flux Normal)
- Matching
- Insulating

Default Boundary Conditions for a Transient Solution

By default, the Boundary/Source Manager automatically assigns the following boundary conditions for a 3D transient model:

Natural boundaries are assigned to the surface between objects.

Neumann boundary conditions (homogeneous) are assigned to the outside faces of the problem region.

Natural Boundary Conditions for a Transient Solution

Natural boundary conditions behave the same way as for magnetostatic or eddy (AC) problems: The normal component of B and the tangential component of H (surfaces without superficial current distribution) are continuous.

Neumann Boundary Conditions for a Transient Solution

Boundaries with Neumann conditions force the magnetic field (H) to be tangential, and the magnetic field (and the associated energy of the field is assumed to be confined to the region with overall Neumann boundaries. Typically, Neumann boundaries are placed at sufficient distance from the sources of the field to avoid an over-constraint of the solution.

By default, the Boundary/Source Manager automatically applies the Neumann boundary to all faces of the problem region — that is, surfaces exposed to the non-meshed space.

Symmetry for a Transient Solution

Symmetry boundary conditions take advantage of the symmetry of a problem. When deciding upon the symmetry, you must consider the geometry, as well as material properties and source distribution in space. Using symmetry boundary conditions when possible is very useful, particularly in 3D transient applications, which typically require large computing resources. Two types of symmetry are available:

- Odd Symmetry (Magnetic Flux Tangential)
- Even Symmetry (Magnetic Flux Normal)

These boundary conditions can be assigned to the faces of the problem region.

Odd Symmetry (Magnetic Flux Tangential) for a Transient Solution

Use an odd symmetry boundary to model a plane of symmetry in a symmetric problem in which current on one side of the plane flows in the opposite direction than current on the other side of the plane.

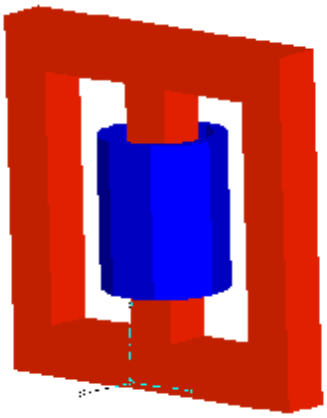
Note	If the odd symmetry plane cuts through an object with eddy currents that tend to flow normal to the magnetic flux tangent boundary, then the odd symmetry condition must be explicitly applied to the model. In this situation, you cannot rely on the default boundary condition.

Even Symmetry (Magnetic Flux Normal) for a Transient Solution

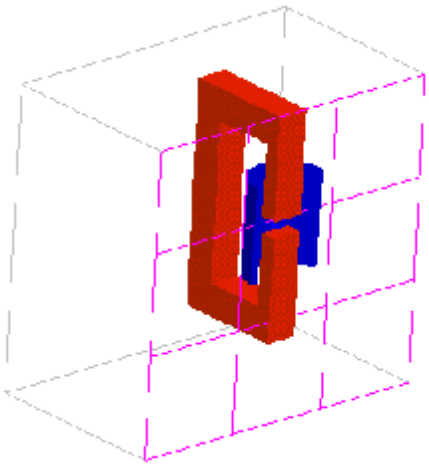
Use an even boundary to model a plane of symmetry in a symmetric problem in which the direction of current flow on both sides of the plane is the same. The currents on both sides of the plane are also assumed to have the same variation in time.

Examples of symmetry are shown in the following four pictures, where the inductor model presented has three-fold symmetry:

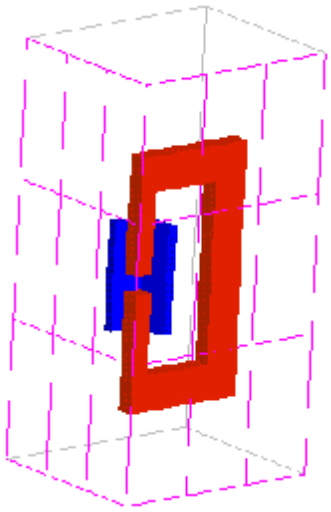
Full Model



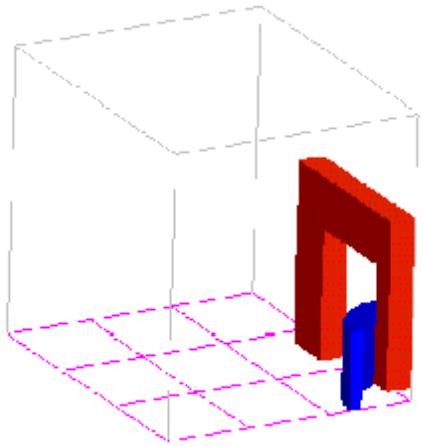
Half Model, Magnetic flux tangent Boundary condition on the symmetry plane



Quarter Model, 2 magnetic flux tangent boundary conditions



1/8 Model, 2 magnetic flux tangent boundary conditions (as in 1/4 model), one magnetic flux normal boundary condition (bottom plane)



Matching Boundary for a Transient Solution

Matching boundaries allow you to model planes of periodicity where the **H** field on one surface exactly matches the **H** field on another by forcing the magnetic field at each location on one surface (the "dependent" boundary) to match the magnetic field at the corresponding location on the other surface (the "independent" boundary). Matching boundaries are used in periodic structures and decrease the resources used in the computational process.

For matching boundaries, you need to set up both an independent and a dependent boundary. Unlike symmetry boundaries on independent and dependent boundaries, the **H** field does not need to be either tangential or normal to these boundaries. However, the **H** field on the two boundaries must have the same magnitude and direction (or the same magnitude and opposite direction) at each time step. The variation in time of the fields at corresponding locations is the same on matching (independent and dependent) boundaries.

Insulating Boundary for a Transient Solution

An insulating boundary prevents current from flowing across a surface - for instance, the interface between two adjacent conductors. Use an insulating boundary to model very thin layers of insulating material between conductors. Modeling thin insulating sheets saves you from the complications associated with the geometry of the insulation and also speeds up the solution process.

Meshing and Band Setting Recommendations for 3D Transient Applications With Motion

Maxwell supports three types of 3D transient problems with motion:

- Translational motion (motion along a user specified, linear direction).
- Rotational motion (non-cylindrical such as the pivoting rotation around an axis encountered in the armature of a relay).
- Rotational motion (cylindrical, such as the type of rotation encountered in an electric machine type of application).

Regardless of the particular type of motion involved, all types of motion applications require a band object that must contain the moving part(s). If there are multiple moving objects, all of the moving parts must be included in one all-inclusive object — this is because they all must be moving as one rigid body, with a single force acting on the assembly. Also, regardless of the type of motion, the user must create a mesh density capable of capturing the physical effects characteristic for the specific application, such as field gradients, skin and proximity effects, etc.

In terms of meshing, the main characteristic of the first two types of motion is that, during the analysis, a new mesh is created at each time step as the motion occurs. This re-meshed region is always inside the band object between the moving part(s) and the stationary part(s). The mesh corresponding to the moving part(s) and the stationary part(s) is kept fixed during the analysis. The mesh density inside the band object is always controlled during the solution process: the edges of any element created inside the band object during re-meshing will never be larger than the average element edge on the entire surface of the band object where it is in contact with both the moving and stationary parts. Thus, both the mesh density created for the moving part (in the area where it touches the band) and the mesh density in the stationary part (in the area where it touches the band) are used to actively control the mesh density in the re-meshed part throughout the entire solution process. This ensures good solution quality for any aspects conditioned by the quality of the mesh.

In maximizing the quality of the solution for 3D transient applications with motion, a number of observations and recommendations are available regarding the band object.

For applications with translational and non-cylindrical rotation types of motion:

- The band object can touch the symmetry plane if any exists.
- The moving object cannot touch a stationary object during motion (the gap between the moving object and the band can never become zero during analysis). The only exception to this rule is for the particular case when both the band and moving object surfaces touch a symmetry plane (if any exists in the respective setup) and the moving object slides along

the symmetry plane (in this case the symmetry plane, respective band, and moving object surfaces coincide);

- The band cannot have true surface faces; all faces must be segmented (for example created with the regular polyhedron primitive).
- The band object cannot separate the stationary part into unconnected sub-regions.

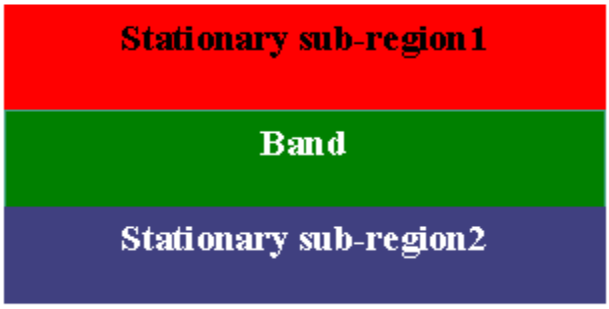


Fig.1 Band cannot separate the stationary part into unconnected regions

- Hollow objects cannot be used for band objects.
- Subtractions cannot be performed within the band object.

For the cylindrical type of rotational motion applications:

For this type of applications the band object must by somewhat larger than the rotating part(s) in ALL directions with only one exception, at the boundaries, which can just be touched. Thus the band object must effectively enclose the moving part, not just by touching it, in the radial and axial directions; so users have to leave some "room" between band and surrounding objects. The only exceptions allowed to the requirement mentioned above are: symmetry plane (if any exists), M-S or other boundaries as applicable for the application, can be touched i.e. at those planes it is allowed for the band surface(s) to coincide with the respective planes.

Additionally, the following observations are also applicable:

- For the band object, always use a facetted (regular polyhedron) type of cylindrical object or a wedge object if symmetry is used. The angular aperture of each facet depends on the problem; however, an opening of 2-3 degrees per facet is usually sufficient.
- Hollow cylinders cannot be used for band objects.
- Subtractions cannot be performed within the band object.
- The band object can separate the stationary part into unconnected sub-regions (a rotor sandwiched between two stators is allowed).

- Non-cylindrical band objects are allowed.

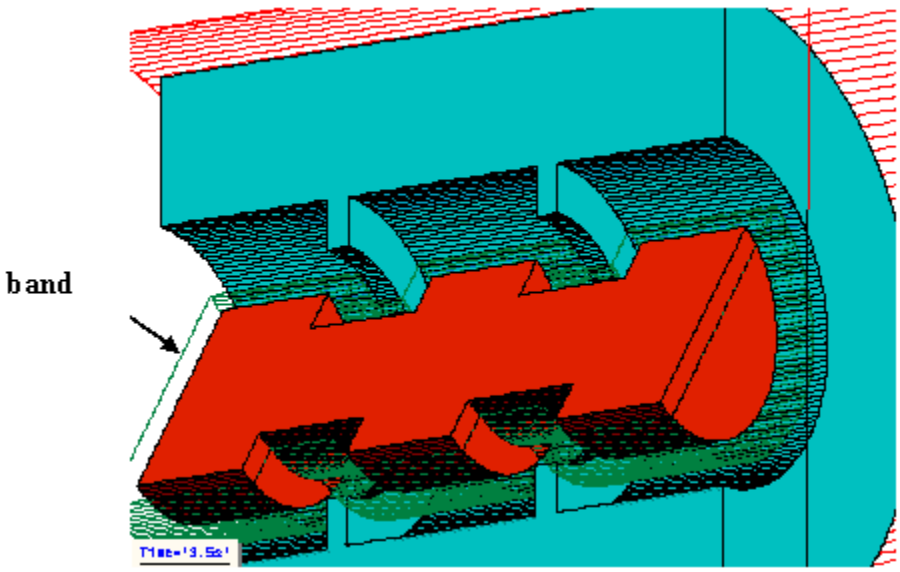


Fig. 2 Rotor and stator assembly with band object in between

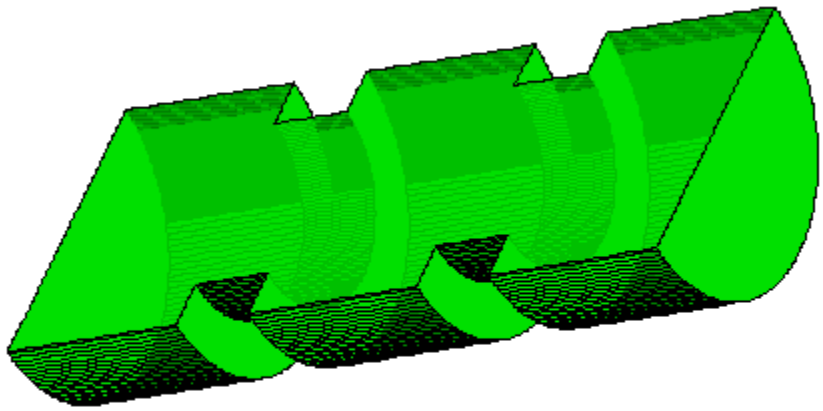
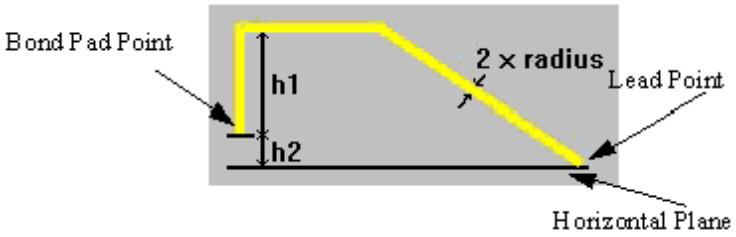


Fig. 3 Band object

Geometric Objects — Bondwires

A bondwire is a thin metal wire that connects a metal signal trace with a chip. You can choose to draw a standard JEDEC 4-point bondwire, as shown below:



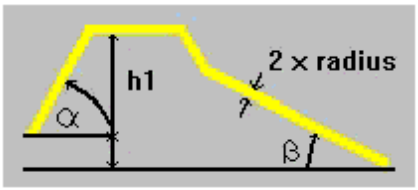
where

h1 = the height between the bond pad point and the top of the loop.

h2 = the height between the lead point and the bond pad point.

radius = half the diameter, or thickness of the wire.

You can draw a standard JEDEC 5-point bondwire, as shown below:



where

α = the angle between the horizontal plane and the wire at the bond pad point.

β = the angle between the horizontal plane and the wire at the lead point.

When drawing the bondwire, first select the bond pad point, a point in 3D space that defines the bond pad position in a horizontal plane. Then select the lead point, which indicates the distance the wire covers in the horizontal plane. Maxwell uses the distance between the bond pad and lead points to calculate the height between the bond pad and the lead point, or *h2*, a value that you can modify in the **Bondwires** dialog box.

Related Topics

Drawing Bondwires

Partial Simulation for Full Rotational Machine Models

A rotational machine can be simulated by using either a 360 deg full model, or a partial model with matching boundary. Full model simulation has higher memory and time cost than partial model simulation; while partial model would require manual operations to cut the model and assign a matching boundary. The manual cut on the geometry could introduce geometry issues which lead to poor mesh quality near matching boundaries.

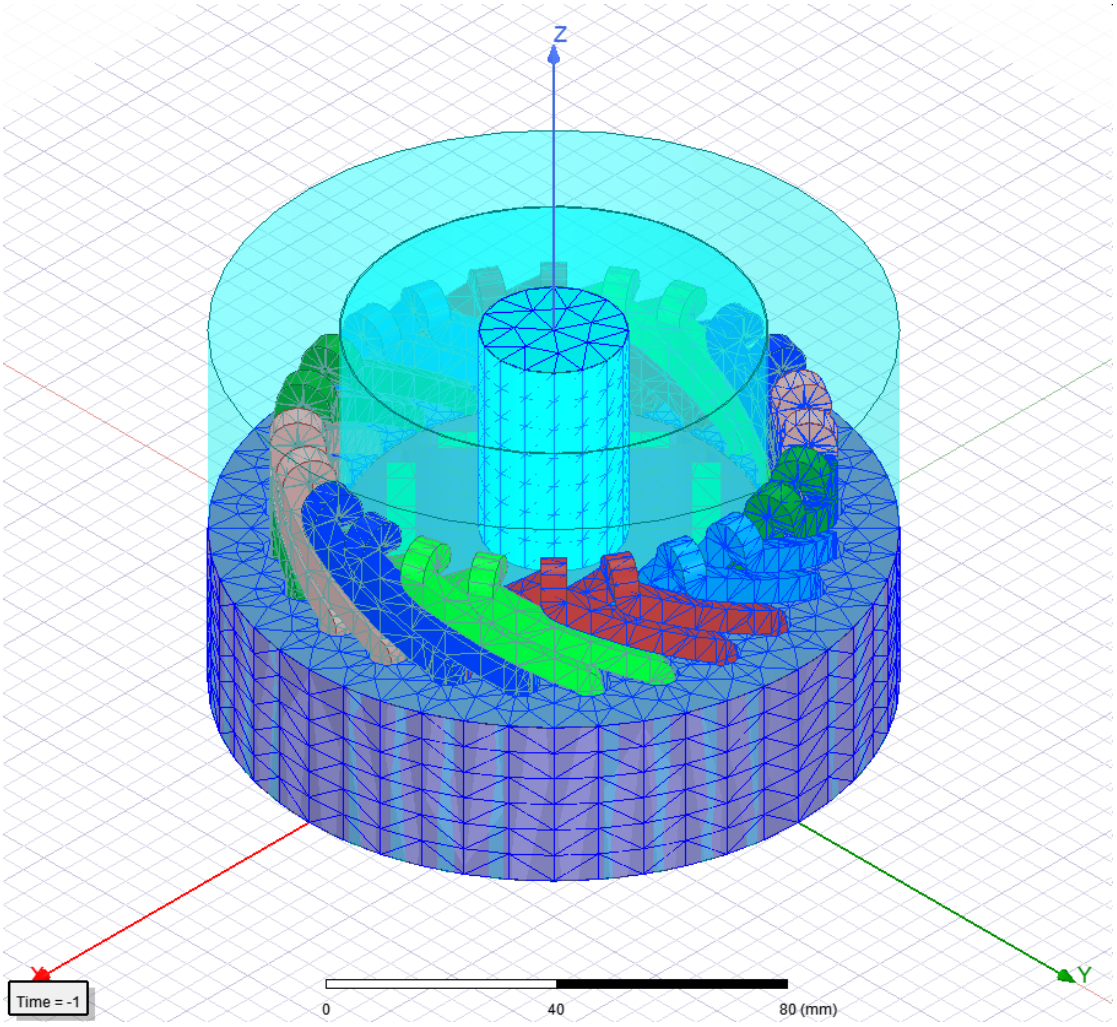
Partial simulation for full rotational machine models is designed to combine the benefits of full model simulation and partial model simulations. Without any manual cutting of the geometry, Maxwell can perform partial simulation for full rotational machines.

Background Information

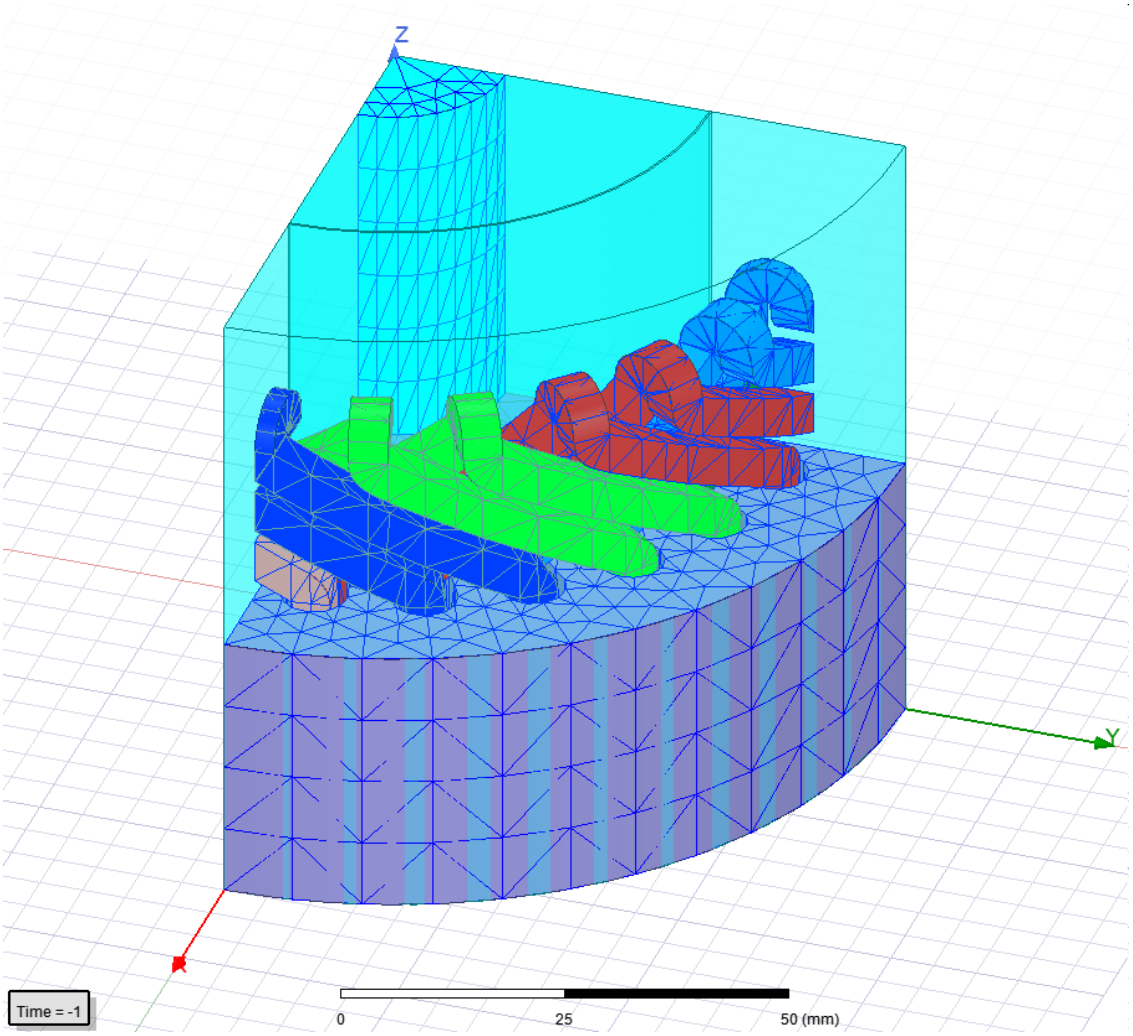
Partial simulation for a full rotational machine model generates and solves a partial mesh on the full model. Instead of a manual cutting of the geometry, this feature allows the mesher to decide where to make the cut. The mesher will also automatically create virtual matching boundaries for the partial mesh.

The following example uses the **ipm_2.aedt** model in the *[install_dir]***Examples\RMxprt\ipm** folder to generate a Maxwell 3D model with **Total circumferential fractions in created 360 deg geometry** on the **Symmetry Multiplier** tab set to 4.

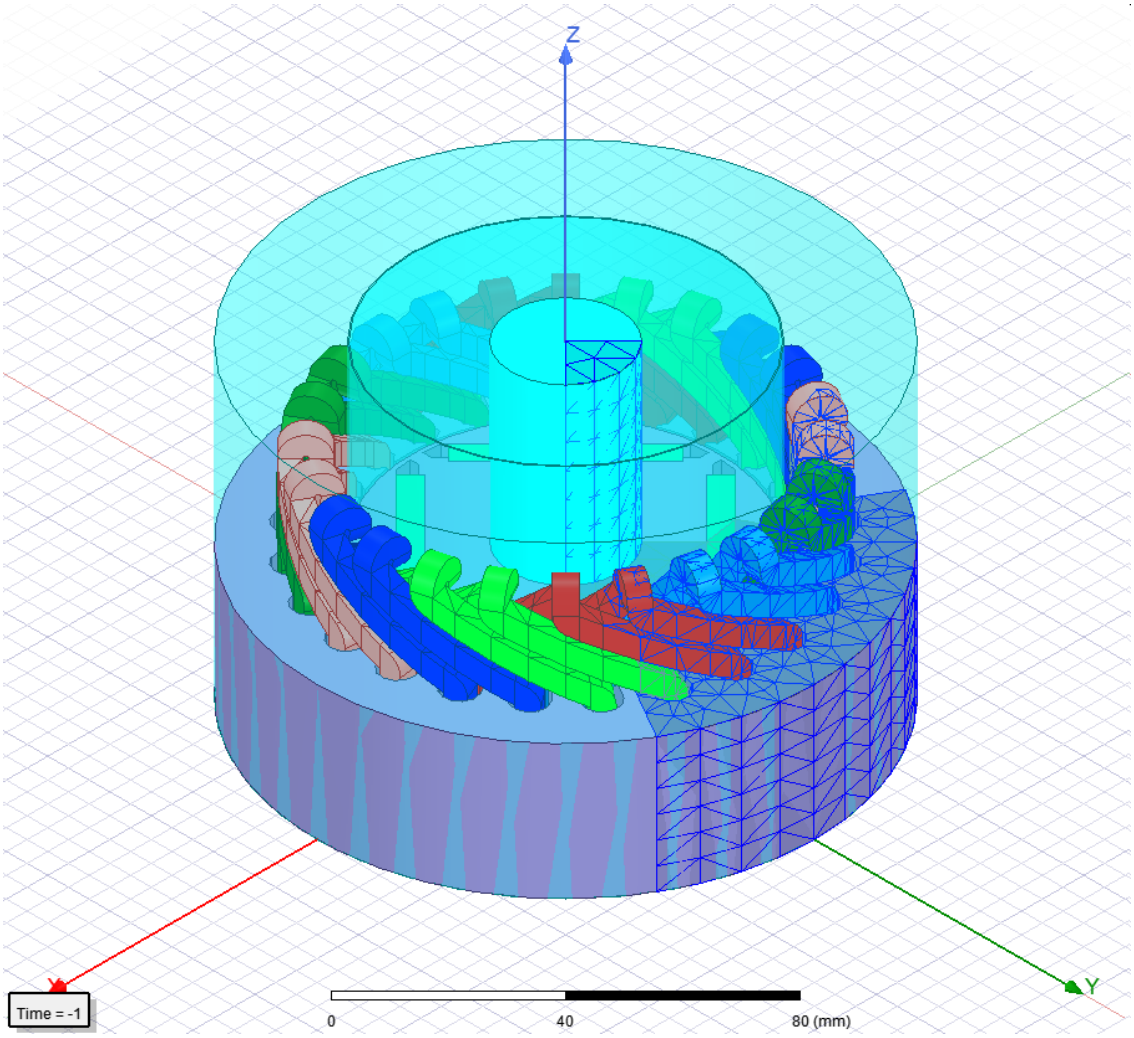
The figure below shows a full mesh on the full ipm_2 model with an element count of 95759.



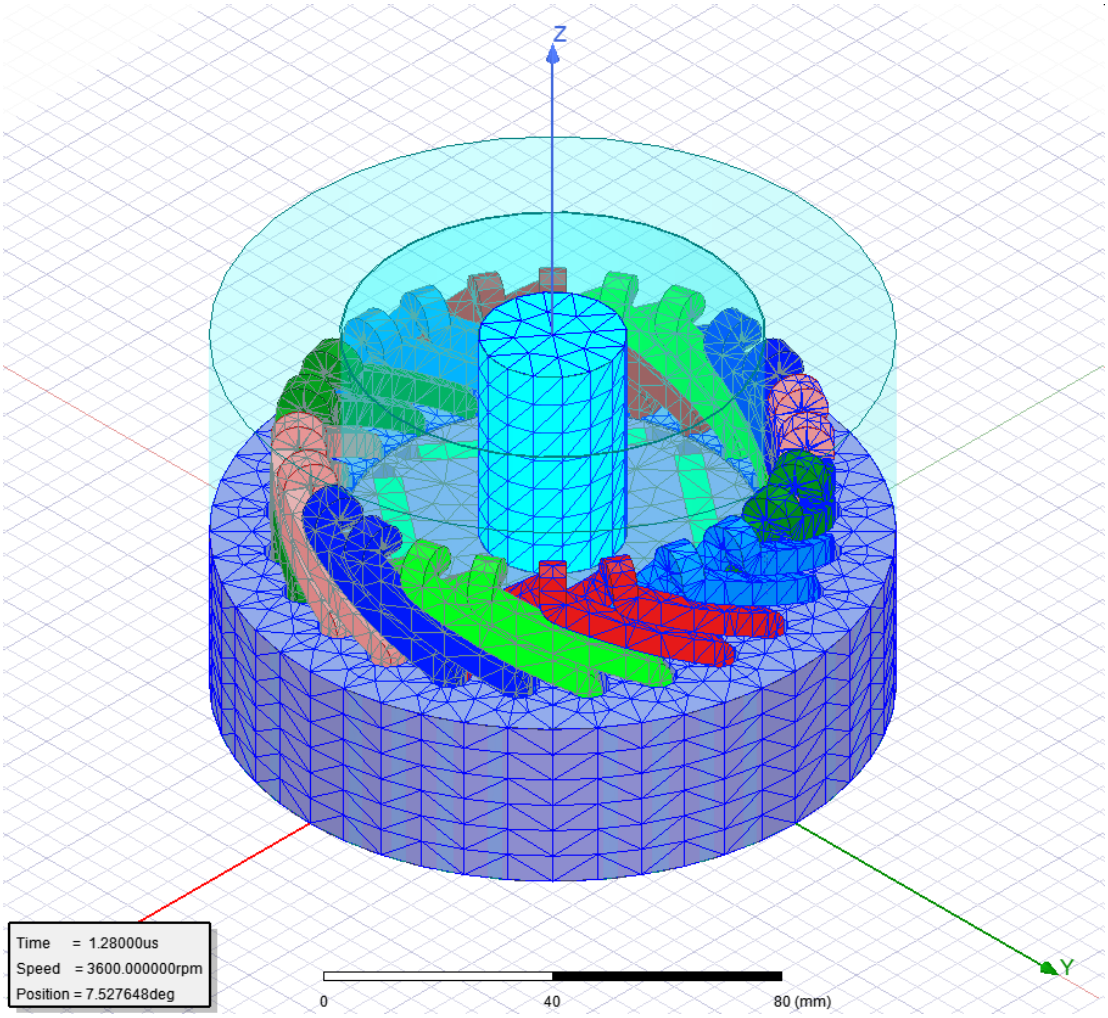
The following figure shows a partial mesh on a manually cut partial model for which the element count is 22194



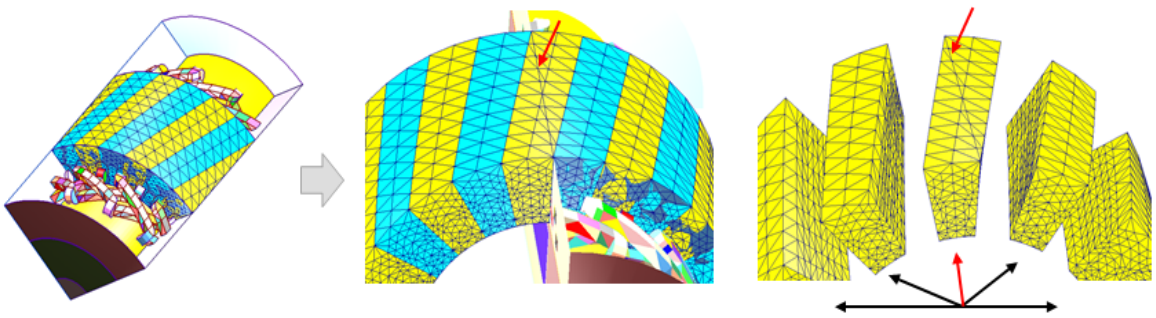
The following figure shows a behind-the-scene partial mesh on a full model for which the element count is 20298. Note that the element count for the partial mesh on a full model is comparable to partial mesh on a partial model, for which neither a manual model cut, nor matching boundary assignment is required.



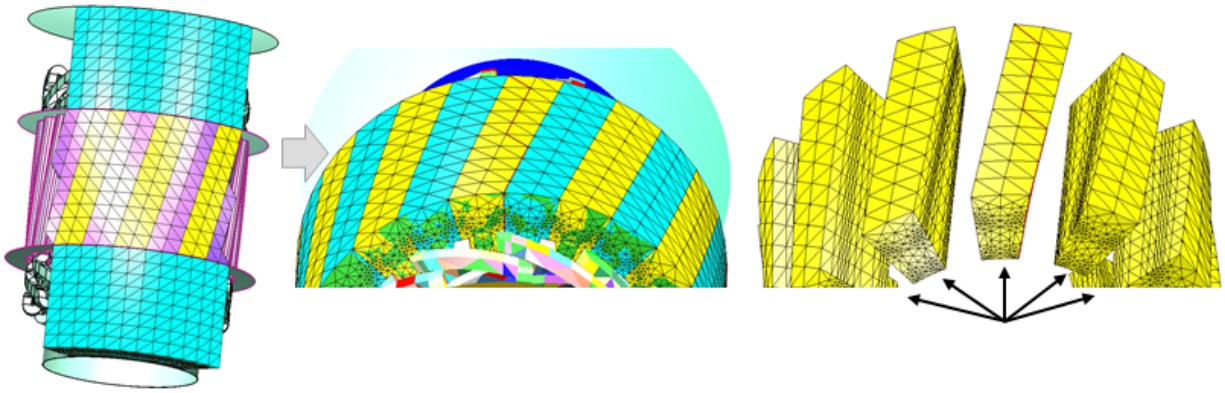
Note that when plotting the partial mesh on the full model, although behind-the-scene mesh only covers part of the model, the mesh plot will be copied and cover the full 360 degree model. Thus, the following model will actually be displayed.



A model with a skewed stator gets even more benefits with partial clone mesh. As shown below, because of the manual cut on the geometry, the skewed model cannot get a perfectly repeatable clone mesh near the matching boundary with partial mesh for a partial model.



On the other hand, with non-planar virtual matching boundary in partial mesh for the full model, identical mesh can be generated for all repeatable geometry patterns.



Workflow for Partial Simulation for a Full Rotational Machine Model

1. After solving the RMxprt machine design, create a full rotational Maxwell 3D machine design with 360 deg geometry.
2. Select **Solve one circumferential fraction** in Symmetry Multiplier settings.
    a. Enter an integer value for the **Total circumferential fractions in created 360 deg geometry**. For example, if a full model can be cut into a 1/4 partial model and matching boundary can be assign at 0 deg and 90 deg of the model, the integer value entered should be 4.
    b. Select either **Full** or **Half** for **Create geometry in axial direction** depending on whether the created model is the half model or the full model in axial direction.
    c. For a rotating machine, the simulation solution is periodic in the circumferential direction. Thus, select either Periodic or Half-periodic for **Field in one fraction** depending on whether the simulation solution in one fraction is of one period or half period physically.
    d. Click **OK** to close the dialog
3. Run simulation on the model.

Related Topics

Running an RMxprt Simulation

Creating a Maxwell Design from RMxprt

Setting a Symmetry Multiplier

Healing Models

> **Note:** Ansys Electronics Desktop (AEDT) uses a third-party modeling kernel to represent 3D geometry and to implement geometric operations in the 3D Modeler. In 2023 R1, AEDT has moved from the ACIS modeling kernel to Parasolid. This is part of an Ansys corporate migration of modeling technologies, based on the strength of the Parasolid kernel, its use by many CAD and CAE applications, and its active maintenance and enhancement by parent company Siemens.

The underlying solid modeling technology used by Ansys Electromagnetics products is provided by the Parasolid kernel geometric modeler. Users can create directly models using primitives and operations on primitives. In addition, users can import models saved in a variety of formats (Step, IGES, etc.) All the models are stored internally in Parasolid native format. When users import models into Ansys Electromagnetics products, translators are invoked that convert the models to this format. Parasolid regularly upgrades their solid modeling system. Third party vendors also create files in Parasolid format but the data in the files is in some cases not robust.

All of the above contribute to errors when a model is read. When translating from (for example) IGES, some accuracy is lost. Not all third party vendors write files to the supported formats with a high level of accuracy. Third party vendors sometimes create invalid files – the files are either targeted at earlier versions of the kernel, or incorporate invalid elements in the file. Sometimes there are errors in reading a file generated by an earlier version with the current version.

The modeling kernel is a highly specialized software component underlying the 3D Modeler in AEDT. The kernel represents the fundamental geometry and implements the operations that create and modify it. Simple geometry and operations are likely to be identical in different kernels and will migrate cleanly. More complex models may have discrepancies due to differences in geometry representation, tolerances, algorithms, or other implementation details. These differences may be subtle, but they can lead to challenges in the migration process.

After migrating to Parasolid, some objects may have invalid geometry or topology, leading to validation or meshing failures. In this case, customers are encouraged to take advantage of the 3D Modeler's built-in healing commands. The Parasolid kernel has very strong healing algorithms and it is often possible to automatically repair such geometry. In the worst-case scenario where healing is not successful, it may be necessary to redraw the object natively in 2023 R1.

Related Topics

Technical Notes: Error Types

Technical Notes: Detecting Errors

Technical Notes: Healing and Meshing

Analyzing Objects

Analyzing Interobject Misalignment

Analyzing Surface Mesh

Healing an Imported Object

Validating Projects

Error Types

When models are imported there are two types of errors – geometry errors and topology errors. Geometry errors are errors in definition of the underlying geometry while topology errors are errors in how the underlying components like faces, edges and vertices are connected. These have to be fixed before mesh analysis can be performed.

When models pass the initial validity checks, mesh generation could still fail. If bodies in the model overlap, mesh will not be invoked. If bodies are very close to each other, mesh might fail. Small features (small edges, small edges, sliver edges) might be present in the bodies which might cause mesh to fail.

The following are common errors present in models:-

1. api_check_entity() errors. These are errors detected by the kernel and are geometry and topology errors.
2. non-manifold topology. These are non-manifold edges and vertices that are present in the model.

Note	One of the requirements for valid mesh generation is that the surface mesh must be ***manifold***, that is, a plot of all the surface triangles of a 3D solid object should not have any “holes.” Such a mesh is often referred to as a ***watertight*** mesh. If there is a hole and you were to fill the object with water, the water would leak out of that hole – the mesh would not be ***watertight***. Zero-area holes (often called cracks) will also “leak.” Having multiple triangles cover the same patch also results in a mesh that is not manifold. Refer to *Detecting Errors* below for information on detecting and correcting such errors.

3. Body pair intersection. This detects if pairs of bodies intersect.
4. Small feature detection – small edge length, small face area and sliver face detection.
5. Mis-aligned entities detection – detects pairs of faces from bodies that can be aligned to remove interbody intersections. This improves the odds of mesh success.
6. Mesh failure error display. This is available for single body, body pairs and last simulation run (all bodies in model). Errors reported by the meshing module are reported to the user.

Errors of type 1, 2, and 3 must be resolved before the mesh can be applied to the model.

Detecting Errors

Errors in the model can be detected by the following methods:

1. Perform a validation check (**Maxwell3D >Validation check**). This returns the following types of errors — api_check_entity() errors, non-manifold errors, and intersection errors — and works on all the bodies present in the model.

2. Run an object analysis (**Modeler >Model Analysis >Analyze Objects**) on a subset of bodies in the model. This returns api_check_entity() errors and non-manifold edge/vertex errors, and, optionally small edge, small face and sliver face detection errors.
3. Run an interobject misalignment analysis (**Modeler >Model Analysis >Analyze Interobject Misalignment**). This yields face pairs from different bodies that are slightly misaligned with respect to each other. This misalignment causes the faces to intersect each other or to create small volumetric portion between the bodies that might cause mesh to fail. If these faces are aligned — that is, the faces are made to share the same surface definition — then the gap between the faces is eliminated and there is a higher likelihood that the mesh will succeed.
4. Run a surface mesh analysis (**Modeler >Model Analysis >Analyze Surface Mesh**). For the selected bodies, a mesh is invoked on each individual body and, optionally, on body pairs. Errors from running the mesh analysis are displayed, helping you focus on bodies/body pairs that fail meshing. If these are fixed, there is a higher probability that the mesh will succeed.
5. Mesh errors from the last simulation run are displayed.

Errors for 2, 3, 4 and 5 are displayed in the **Model Analysis** dialog box. Using the menu items for 2, 3 and 4 displays the dialog box. This dialog box can also be opened by clicking **Modeler >Model Analysis >Show Analysis Dialog**.

Healing and Meshing

Potential problems with 3D Models

This section lists problems that can prevent a 3D model from being meshed successfully. Subsequent sections will describe how these problems can be detected and addressed.

Kernel errors

The underlying solid modeling technology used by Ansys Electromagnetics 3D products Maxwell, HFSS and Q3D is provided by the Parasolid kernel. You can create models directly in the drawing environment of these Ansys Electromagnetics products using primitives, such as boxes, cylinders, etc. and operations on primitives, such as Boolean operations. In addition, you can import models produced by other CAD tools in a variety of formats such as STEP, IGES, etc. In Ansys Electromagnetics 3D products, all models are stored internally in the kernel's native format. When you import models into Ansys Electromagnetics products, translators are invoked that convert the models to this format. Often, models that were created in other CAD tools were created initially for other purposes than electromagnetic analysis, such as,. for mechanical design or just for display purposes. They may have imperfections that make them illegal to the kernel. Further, there can be compatibility issues between different versions and even flavors of modeling tools. All this can lead to errors in imported 3D models.

If you use Ansys Electromagnetics products to create geometry models, and thereby avoid model import and translation, you are unlikely to encounter such problems.

Mixed dimensionality

Even if a model is imported and translated without errors, there is a restriction to be aware of. The kernel cannot handle mixed-dimensionality models. One of the goals of Ansys Electromagnetics use of the kernel modeling system is to create a valid volumetric mesh for simulation. Mixed-dimensionality models will not yield a valid volumetric mesh. Therefore, the Ansys Electromagnetics tools will not mesh objects with mixed dimensionality, so-called non-manifold objects. For instance, imagine a 3D object representing a curved metal plate with a small but finite thickness. If it reaches zero thickness somewhere while having non-zero thickness elsewhere, it has mixed dimensionality, 2D as well as 3D. You will get an error message saying that the object is non manifold. Of course, 2D and 3D objects can co-exist in a model, but any one object cannot be both 2D and 3D.

Intersecting objects

Another restriction is that Ansys Electromagnetics 3D tools do not allow partial intersections (also known as partial overlaps) between 3D objects. Each element of the mesh has to belong unambiguously to one object. There is no problem if one object is enclosed completely inside a bigger object, but partial intersections lead to ambiguities. As long as there are partial object intersections, the mesh generator will not attempt to create a mesh. Instead, you will get an error message notifying you which objects are intersecting. You must remove the intersections before you can proceed. You can do this by changing the shapes of objects slightly, or by subtracting one object from the other.

Caveat: If, as a result of a subtraction, the model has pairs of true surfaces that are coincident (that is, smooth curved surfaces that fit exactly one inside the other), you make it harder for the mesh generator to create a mesh. This is because the kernel creates segmentations on each of these surfaces, and these segmentations are not guaranteed to fit. Setting a small value for Surface Deviation under ***Mesh>Assign Mesh Operation>Surface Approximation*** *increases your chance of success in such a case, but it is better to avoid such situations if you can.*

Small features and misalignment

When there are no kernel errors in the model, no non-manifold objects and no partial object intersections, the mesh generator can be invoked to create a valid mesh for the electromagnetic analysis. Even if the geometry is valid, mesh generation can still fail. Possible causes are the presence of very short edges, very small faces, long and thin sliver faces, and slight misalignments between faces that are supposed to be coincident.

Related Topics

Technical Notes: *Detecting and Addressing Model Problems to Improve Meshing*

Detecting and Addressing Model Problems to Improve Meshing

The following sections describe a systematic procedure to detect and address model problems that can interfere with the meshing process.

Technical Notes: *Healing During Geometry Import*

Technical Notes: *Healing After Geometry Import*

Technical Notes: *Removing Object Intersections*

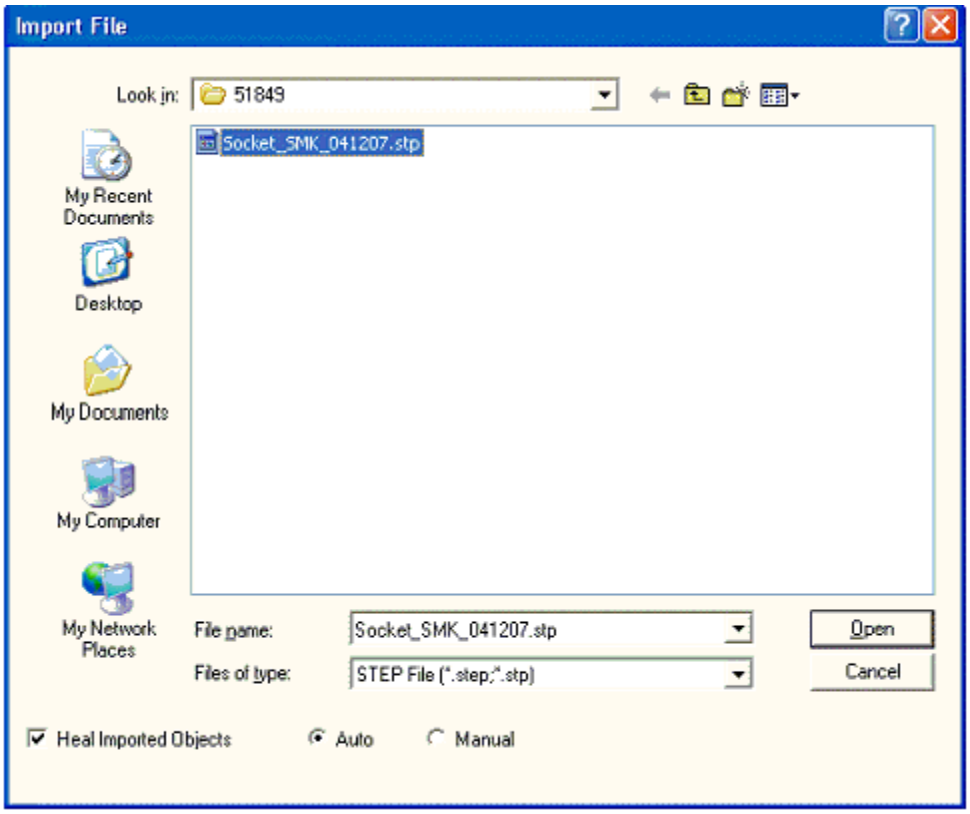
Technical Notes: *Removing Small Features*

Technical Notes: *Aligning Objects*

Technical Notes: *Troubleshooting if Meshing Still Fails*

Stage One: Healing during geometry import

In case you do not draw your entire geometry in the Ansys Electromagnetics environment but wish to import (part of) it, in the **Import File** window you select which geometry file to import. At the bottom of this window is a check box "**Heal Imported Objects**". Two modes exist, "auto" and "manual". Auto Healing will try to address geometry kernel errors and non-manifold errors, the first two classes of potential problems listed earlier. It will also fix surface normals in the body and updating orientation of body, to avoid having a body with negative volume.



Manual healing adds small-feature removal to this. You can remove small features at this stage if you wish. However, the usual approach is to apply auto-healing at this stage and leave small-feature removal until later.

Stage Two: Healing after geometry import

Healing can only be performed on objects that have no drawing history other than "import". If necessary, object history can be deleted through **Modeler >Purge History**. If that causes a warning that another object will be deleted, you may need to purge the history of that other object first, or purge the histories of several objects simultaneously.

At any time after import, you can perform a Validation Check: **Maxwell3D >Validation Check**. This will enable you to focus on bodies and body pairs that need attention before a mesh can be created.

1. Select the objects that have geometry kernel errors, such as failing api_check_entity(), and the objects that have non-manifold features, i.e. mixed dimensionality.
2. Invoke **Modeler >Model Analysis >Analyze Objects**.

   This will bring up an **Options** dialog to set thresholds for small feature detection and on completion, the **Model Analysis** dialog is displayed. All bodies in the model are shown in the objects grid along with their status. Bodies can have the following status:

1. Good
2. Null Body
3. Analysis not performed
4. Invalid entities found
5. Small-entity errors

Invalid-entity errors are geometry kernel errors and non-manifold errors. Small-entity errors are small faces, sliver faces and small edges that are optionally detected based on user-defined parameters.

Note	Invalid-entity errors must be fixed before a mesh can be generated.

To fix invalid entity errors:

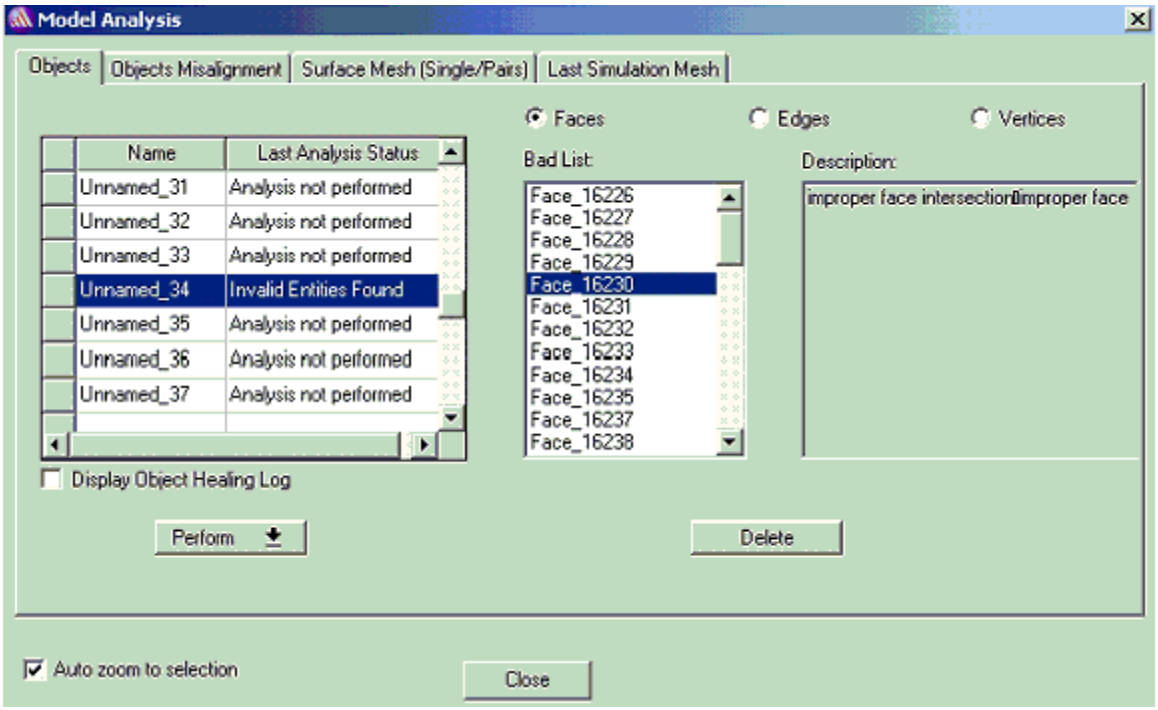
1. Choose the bodies that have "Invalid Entities Found."
2. In the same window, choose **Perform->Heal Objects**, with or without an optional setting for small-feature removal.

   In most cases, the bodies will be healed and the errors fixed.
3. If errors still persist, choose "offending" faces and edges and click on **Delete**.

   This will replace the selected face/edge entity by a tolerant edge/vertex respectively.

In order to avoid unintended changes, it is good practice to do the following:

1. At the bottom of the **Model Analysis** window, check the box "Auto Zoom to Selection."
2. Select one face or edge at a time
3. Decide for each face and edge whether you want to delete it.



Note	Healing causes changes to the geometry and topology of the body being healed. Validation check has to be re-run after healing is done to identify body pairs that intersect. It is possible that after healing, bodies that were disjoint before now overlap.

In some cases the replacement of the face/edge by tolerant edge/vertex will fail. If the object remains invalid, you know at this point what parts of the object are invalid. You will need to change that part of the object manually, either in Ansys Electromagnetics drawing environment or in the original CAD tool, to make it pass. Often, the invalid entities are in small details that can be changed without noticeably affecting the results of the electromagnetic analysis. For example, it may be possible to create a small object, well placed in the "offending" region, and to unite it with or subtract it from the problematic object, such that the "offending" details no longer exist.

Stage Three: Removing Object Intersections

If there are any intersecting objects, a Validation Check will list them. You must eliminate object intersections before a mesh can be created.

In complicated models, before making changes, it is good practice to inspect the overlap visually. A way to do this is to:

1. Duplicate both objects and place the copies outside the model.
2. Perform Boolean Intersect on the copies.

   This will show you what causes the intersection and will help you decide how to remove it.
3. Then, delete the copies.

The easiest way to eliminate object overlap is to subtract one object from the other, in the order that leaves the desired material in the region of overlap. If the overlap is very small and you can choose the order of subtraction, choose one that does not create coincident true surfaces, if possible.

Caveat: if as a result of a subtraction the model has pairs of true surfaces that are coincident, that is, smooth curved surfaces that fit exactly one inside the other, you will make it harder for the mesh generator to create a mesh. This is because the geometry kernel will create segmentations on each of these surfaces, and these segmentations are not guaranteed to fit. Setting a small value for **Surface Deviation** under **Mesh>Assign Mesh Operation>Surface Approximation** increases your chance of success in such a case, but it's better to avoid such situations if you can.

A way to eliminate object intersections without subtraction is to split one object in parts, in such a way that some parts are completely enclosed in the other object, and some parts are completely outside the other object. Even for complicated objects, this is possible through a sequence of Boolean operations on the objects and copies of the objects.

At this point, the geometry has no geometry kernel errors, no non-manifold objects and no partial object intersections. A mesh can be created for the electromagnetic analysis.

Stage Four: Removing Small Features

Even though, in principle, the geometry may be ready for a mesh to be created, it is possible that small features in the geometry lead to a mesh that is unnecessarily large and contains long and thin tetrahedra that make the simulation converge slower. Small features may even cause the mesh generation to fail. By small, we mean details on an object that are thousands of times smaller than the main features of the object, and that, in most cases, are unintended consequences of the drawing history in another CAD tool. Therefore, it is advantageous to remove small features.

To do this, you may need to purge the history of objects, since healing and related operations can only be performed on objects without history beyond import. You may have noticed that you could have invoked small-feature removal at several earlier stages. There is no objection to doing it earlier. The reason why it is presented here as stage four is that the previous stages were necessary while this one is optional.

To start the small-feature removal:

1. Select objects and invoke object analysis through **Modeler >Model Analysis >Analyze Objects**.

   Alternatively, without objects selected, use **Modeler >Model Analysis >Show Analysis Dialog >Objects** and select objects from the list. In the **Model Analysis window**, invoke **Perform >Analyze Objects**.

   The software will report the smallest edge length and the smallest face area, and enable you to set thresholds for the detection of short edges, small faces and sliver faces.
2. Upon clicking OK, the analysis is performed.

   As a result of the analysis, the software presents a list of all faces and edges that do not meet the thresholds set by you.

3. Check the box "Auto-Zoom to Selection" at the bottom of the Model Analysis window and click on small faces and short edges in the list.

   Inspect them visually and decide whether they can be deleted. It is good practice to delete them one by one rather than deleting many at once in order to prevent unintended changes.

   Sometimes, an edge or face cannot be deleted, and you get a message notifying you. In that case, either ignore it, or revisit it after deleting some other details first, or revisit it later manually in the 3D drawing environment.

   At this point, the geometry has no errors, no non-manifold objects and no partial object intersections. Furthermore, there are fewer small features that were unintended or unimportant for the electromagnetic analysis, so the quality of the model has improved.

Stage Five: Aligning Objects

Objects that touch each other in imported geometries do not always have well-aligned faces. Often, this is a consequence of the limited level of precision in the imported file. Misaligned faces can cause tiny object intersections or tiny gaps between objects, which in turn can lead to an inefficient mesh or even a failure to create the mesh.

To repair such occurrences in an automated way, you can select groups of objects and invoke **Modeler >Model Analysis >Analyze Interobject Misalignment**. This will yield face pairs from different bodies that are slightly misaligned with respect to each other.

In the window that shows this list, check the box "**Auto-Zoom to Selection**" and select face pairs from the list. When you decide that faces should be aligned, click Align Faces. In some cases, face alignment will fail if the topology of the body would change by a large amount after alignment. In that case, you can decide to ignore it, as it may not be a problem, or revisit it later manually in the 3D Modeler environment.

Note	In complicated models, the **Interobject Misalignment** analysis can take a long time if you select all objects before launching the analysis. If you do not know which pairs of objects to analyze, just let the mesh generator try to make a mesh. If the mesh fails, a list is presented to you of misalignments that the mesh generator finds suspicious but did not want to adjust without permission. Not every misalignment in the list is always a problem: this is a list of features that might need your attention.

Note	As face misalignments between touching objects can cause small object intersections, this alignment capability can already serve a useful role in stage three.

Stage Six: Troubleshooting if meshing still fails

If mesh generation fails, information about the reasons for the failure is presented under **Modeler >Model Analysis >Show Analysis Dialog >View Mesh Feedback**.

Again, check the box "Auto Zoom to Selection" and click on the errors in the list. This can give you hints about which parts of the model are causing difficulties.

For instance, there may be self-intersecting bodies or faces. Such errors can have a variety of causes, such as a face that is supposed to be planar, but of which the vertices do not quite lie in the same plane. When you zoom and search you are likely to see what causes the problem.

Also, there may be face misalignments. Once you know they exist, you can inspect them and decide whether to align them under the Objects Misalignment tab.

One of the tabs of the **Model Analysis** window is the **Surface Mesh** tab. Under that tab, you can try to create surface meshes for objects and pairs of objects. Since a surface mesh on selected objects is easier to create than a volume mesh for the whole model, this can help you to identify quickly which objects are causing difficulties and why.

Also, in order to determine which objects are causing difficulties, you can exclude objects temporarily from the model. If the mesh succeeds without them, this helps to identify the reason for failure. To exclude an object temporarily, select it and uncheck "Model" in its properties window. Then try to create the mesh again.

Once you know which objects make the mesh fail, you can try to make small changes to them that do not affect the electrical properties noticeably but help the mesh maker succeed. For example:

- Zoom in on details and consider removing details;
- Find coincident true surfaces and move one of the faces over a very short distance so the pair of faces is not coincident anymore;
- Split very complicated objects into multiple less-complicated objects;
- Delete a complicated 3D ground object and create a 2D ground through a boundary condition on the appropriate faces of a dielectric;
- Replace imported objects by objects drawn in Ansys Electromagnetics 3D modeling environment. For instance, some CAD tools produce cylinders that consist of two half cylinders that have a seam where they join. The fit is not always perfect.

Finally, for coincident true surfaces, set a very small value for **Surface Deviation** under **Mesh> Assign Mesh Operation> Surface Approximation**. Parasolid will give them more segments, but you can compensate for that with **Model Resolution**. In a parametric sweep, you can experiment with settings for **Surface Deviation** and **Model Resolution**.

Related Topics

Analyze Objects

Analyze Interobject Misalignment

Analyze Surface Mesh

Healing

Validating Projects

Handling Complicated Models

Complicated models, often imported from a CAD tool or layout tool, may slow down the interface, use a lot of RAM during file I/O and other operations, contain imperfections and object overlaps.

After analysis, post processing of such models may be time consuming. Maxwell has several options and features that address these problems.

- Interface Options for Complicated Models
- Geometry Imperfections and Complicated Models
- Object Overlap Settings for Complicated Models
- Post Processing Settings for Complicated Models

Interface Options for Complicated Models

To improve the speed of the interface when dealing with complicated geometries, do the following:

Under **Tools>Options>Modeler Options**, on the **Display** tab...

- Set "Default View Render" to "Wire Frame". Wire-frame rendering is faster than shaded rendering.
- Turn off "Display UV Isolines". For models with curved faces, this will simplify the wire-frame display, so the rendering will be faster.
- Turn off "Visualize History of Objects". This will remove visualization of objects that are part of the model history. For large models, this is faster and uses less memory.

Under **View>Visualization Settings**

- Use larger deviations to view curved objects in less detail.

Under **Modeler>Import**

- Uncheck "Check Model" and "Heal Imported Objects." This helps for complicated models: Validation and healing take considerable time for such models. Use this option to defer checking to a later stage (especially in cases where you know that you want to mesh the model as is).

Under **Tools>Options>General Options**, on the **Project Options** tab…

- Turn off "Do Autosave" or set the autosave interval to a larger value, for example, 50. Auto-save can be time consuming.

Geometry Imperfections and Complicated Models

Modeler>Import

Many formats can be handled. It is recommended to import a version of the geometry that is as close as possible to its source, rather than geometries that have been translated before from one format to another, or that have been imported into another computational tool and later exported from it.

Modeler>Validation Settings

Geometry imperfections are listed as kernel errors when executing **Maxwell>Validation Check** and when starting an analysis. It is recommended to attempt to heal objects with such errors. However, Maxwell enables you to bypass the errors (not the check itself) by choosing a setting under **Modeler>Validation Settings**. "Warning Only" enables you to ignore all errors. "Basic" enables you to bypass all but the most severe errors. The Maxwell mesh generator has been enhanced to handle many geometry errors.

Object Overlap Settings for Complicated Models

In the **Set Material Override Tab** in the **Maxwell3D>Design Settings** menu...

Complicated geometries often have small object overlaps. This setting will allow overlaps between dielectrics and metals. In the overlap region, the metal will locally take priority over the dielectric, as if this part of the dielectric has been subtracted. Overlaps between two dielectrics and overlaps between two metals are still not allowed.

Post Processing for Complicated Models

Under **Tools>Options>General Options**, on the **Miscellaneous Options** tab...

- Turn off "Dynamically update postprocessing data during edits". This will disable expensive updating of existing reports and plots.
- Turn off "Update reports on file open". This will disable expensive updating of reports and plots when opening a project.

Materials

This section of the *Technical Notes* includes information on the following material properties:

Simple Materials

- Relative Permeability
- Relative Permittivity
- Bulk Conductivity
- Dielectric Loss Tangent
- Magnetic Loss Tangent

Frequency Dependent

- Frequency-dependent material properties.

Anisotropic materials

- Anisotropic materials
- Anisotropic relative permeability tensors
- Anisotropic relative permittivity tensors
- Anisotropic conductivity tensors
- Anisotropic dielectric loss tangent tensors
- Anisotropic magnetic loss tangent tensors
- Anisotropic materials and ports

Ferrite materials

- Magnetic Saturation
- Lande G Factor
- Delta H

Magnetostrictive materials

- Magnetostriction Modeling of Nonlinear Magnetic Material

Relative Complex Permeability

The permeability of dielectrics is assumed to be complex, as follows:

$$\mu = \mu' - j\mu''$$

The relative permeability is a dimensionless quantity, defined as follows:

$$\mu' = \mu_r \mu_0$$

where μ_0 is the permeability of free space.
The permeability can also be expressed as

$$\mu = \mu'\left(1 - j\frac{\mu''}{\mu'}\right)$$

where μ' is the real portion of μ and μ''/μ' is the magnetic loss tangent.

Relative Complex Permittivity

The permittivity of dielectrics is assumed to be complex, as follows:

$$\varepsilon = \varepsilon' - j\varepsilon''$$

which can also be expressed as

$$\varepsilon = \varepsilon'\left(1 - j\frac{\varepsilon''}{\varepsilon'}\right)$$

where ε' is the real portion of ε and from which the relative permittivity is defined as:

$$\varepsilon' = \varepsilon_r \varepsilon_0$$

where ε_0 is the permittivity of free space.

$\varepsilon''/\varepsilon'$ is the dielectric loss tangent.

If a material's losses due to bulk conductivity will be significant, such as in semiconductor dielectric materials, an additional bulk conductivity value, σ, must be added. From the time harmonic form of Maxwell's equations, the complex permittivity, ε_c is defined as

$$\nabla \times H = (\sigma + j\omega\varepsilon)E = j\omega\varepsilon_c E$$

where

$$\varepsilon_c = \varepsilon(1 - j\tan\delta - j\frac{\sigma}{\omega\varepsilon})$$

Bulk Conductivity

Maxwell is capable of including conductivity in the model either as a bulk material loss factor, similar to dielectric loss tangent, or as an impedance boundary condition applied to the outer surfaces of the object. The choice of bulk material loss instead of the boundary condition is made by selecting **Solve Inside** in the **Properties** window and by adding an Impedance Boundary on the surface of the object.

The choice between bulk material loss and the surface impedance boundary condition is problem dependent. The boundary condition can be applied whenever the conductor is much thicker than the skin depth at the solution frequency. In this case, the unknowns within the conductor are not included in the unknown vector, resulting in a smaller matrix and a faster analysis. However, if the conductor is not thick relative to the skin depth, the bulk material conductivity must be used to arrive at an accurate solution. With this assumption, the wave equation reduces to

$$\nabla\times\left(\frac{-1}{j\omega\mu}\nabla\times\boldsymbol{E}\right) = (j\omega\varepsilon + \sigma)\boldsymbol{E}$$

Related Topics

Impedance Boundary

Dielectric Loss Tangent

To represent a dielectric that dissipates the power of a high-frequency electric field, enter a dielectric loss tangent, $\varepsilon''/\varepsilon'$, property value for the material. The smaller the loss tangent, the less

lossy the material.

The dielectric loss tangent may vary with frequency. To simulate the variances, define a function for the dielectric loss tangent.

Magnetic Loss Tangent

To represent a magnetic material such as ferrite that dissipates the power of a high-frequency magnetic field, enter a magnetic loss tangent, μ''/μ', property value for the material. The smaller the loss tangent, the less lossy the material.

Note	If you plan to do a fast frequency sweep for a design that includes dielectrics, make sure that the dielectric or magnetic loss tangent does not vary significantly over the requested frequency range. If they do, the results may not be what you expect. In cases where the loss tangent does vary significantly over the frequency range in which you are interested, copy and solve the design several times, adjusting the loss tangent and associated frequency range for the copied design so that the loss tangent is relatively stable over the design's requested frequency range.

Note	When magnetic loss tangent is used, the resulting hysteresis core loss will be reflected in the resistance term for both the impedance matrix and the Simplorer Dynamic Eddy Current ROM.

Imaginary Permeability

Eddy Current

Some materials exhibit a permeability that includes both a real and imaginary component. The imaginary component is used to model magnetic losses in a time-varying field using the relationship:

$$B = (\mu_r' - j(\mu_r''))\mu_0 H$$

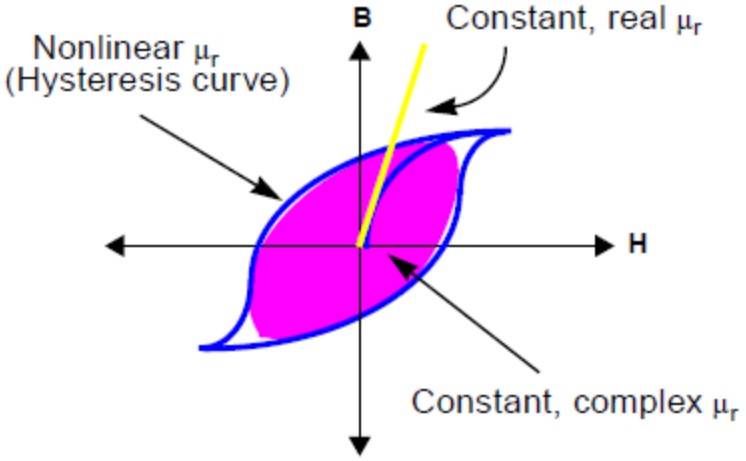
where:

μ_r' is the real component of the relative permeability.

μ_r'' is the imaginary component of the relative permeability.

μ_0 is the permeability of free space.

As shown below, a complex relative permeability causes the B field to lag behind the H field – similar to the behavior of a nonlinear, lossy material. The power loss during this cycle (the color shaded area) is approximately equal to the hysteresis loss (the area within the blue lines). The hysteresis curve for a material with a constant, real permeability (the straight yellow line) is shown as a reference.



When the nature of losses is also due to magnetic hysteresis, Maxwell allows you to model such effects using complex permeability by allowing the specification of the tangent of magnetic loss. When a linear material exhibits hysteresis effects, it should be characterized using a complex permeability, as below:

$$\underline{\mu} = \mu' - j\mu'' = \mu'\left(1 - j\tan\delta\right)$$

where $\tan\delta$ is the magnetic loss tangent, $\frac{\mu''}{\mu'}$

Enter the imaginary relative permeability of a material, *μ''*, in the **Imag. Permeability** field. The default imaginary permeability of zero is that of a material that exhibits no magnetic loss in a time-varying field.

Ferrite Materials

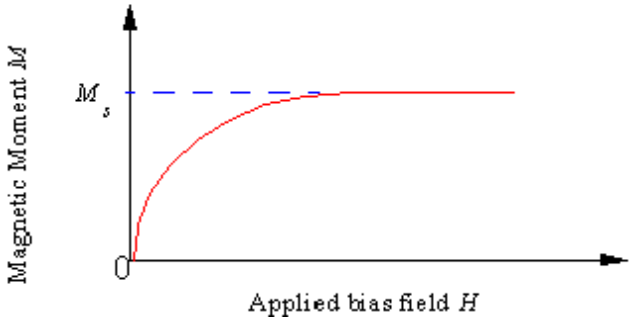
Ferrite materials are used to model the interaction between a microwave signal and a material whose magnetic dipole moments are aligned with an applied bias field. The gyrotropic quality of the ferrite is evident in the permeability tensor which is Hermitian in the lossless case. The Hermitian tensor form leads to the non-reciprocal nature of the devices containing microwave ferrites. If the microwave signal is circularly polarized in the same direction as the precession of the magnetic dipole moments, the signal interacts strongly with the material. When the signal is polarized in the opposite direction to the precession, the interaction will be weaker. Because the interaction between the signal and material depends on the direction of the rotation, the signal propagates through a ferrite material differently in different directions.

If you assign a ferrite material to an object, you must assign a magnetic bias source to the object.

Magnetic Saturation

A material with a non-zero magnetic saturation is considered to be a ferrite. When a ferrite is placed in a uniform magnetic field, the magnetic dipole moments of the material begin to align with the field. As the strength of the applied bias field increases, more of the dipole moments align. The magnetic saturation, M_s, is a property that describes the point at which all of the magnetic dipole moments of the material become aligned. At this point, further increases in the applied bias field

strength do not result in further saturation. The relationship between the magnetic moment, *M* and the applied bias field, *H*, is shown below.



The magnetic saturation, $4\pi M$, is entered in gauss.

Lande G Factor

The Lande *g* factor is a ferrite property that, on a microscopic level, describes the total magnetic moment of the electrons according to the relative contributions of the orbital moment and the spin moment. When the total magnetic moment is due entirely to the orbital moment, *g* is equal to one. When the total magnetic moment is due entirely to the spin moment, *g* is equal to two. For most

microwave ferrite materials, *g* has a range from 1.99 to 2.01. The Lande *g* factor is dimensionless.

Delta H

Delta H is the full resonance line width at half-maximum, which is measured during a ferromagnetic resonance measurement. It relates to how rapidly a precessional mode in the biased ferrite will damp out when the excitation is removed. The factor ΔH doesn't appear in the permeability tensor; instead, the factor α appears. The factor α is computed from

$$\alpha = -\frac{\gamma \mu_0 \Delta H}{2\omega}$$

The factor α changes the κ and χ terms in the permeability tensor from real to complex, which makes the tensor complex non-symmetric (where it had been hermitian for lossless ferrites).

Enter the full resonance line width at half maximum in the **Delta H** value box. Delta H is measured at a specific frequency. That frequency needs to be entered by the user at the interface. 9.4 Ghz is the frequency where Delta H is typically measured.

Anisotropic Materials

Anisotropic materials have characteristics that vary with direction. These characteristics are defined by their anisotropy tensors. You must define three diagonals each for anisotropic permittivity, electric loss tangent, conductivity, permeability, and magnetic loss tangent. Each diagonal represents a tensor of your model along an axis. The alignment of the materials axis with respect to the object is discussed in Assigning Material Property Types.

The following anisotropic characteristics can be defined:

- Anisotropic relative permeability tensors
- Anisotropic relative permittivity tensors
- Anisotropic conductivity tensors
- Anisotropic dielectric loss tangent tensors
- Anisotropic magnetic loss tangent tensors
- Anisotropic materials and ports

Related Topics

Stacking Direction
Lamination Modeling
Assigning Material Property Types
Setting Coordinate Systems
Creating a Relative Coordinate System
Change the Orientation of an object
Relative Permeability
Anisotropic Relative Permeability Tensors
Relative Permittivity
Bulk Conductivity
Dielectric Loss Tangent
Magnetic Loss Tangent

Anisotropic Relative Permeability Tensors

The relative permeability tensor for an anisotropic material is described by

$$[\mu] = \begin{bmatrix} \mu_1\mu_0 & 0 & 0 \\ 0 & \mu_2\mu_0 & 0 \\ 0 & 0 & \mu_3\mu_0 \end{bmatrix}$$

where

- μ_1 is the relative permeability along one axis of the material's permeability tensor.
- μ_2 is the relative permeability along the second axis.
- μ_3 is the relative permeability along the third axis.
- μ_0 is the permeability of free space.

The relationship between *B* and *H* is:

$$\begin{matrix} x \rightarrow \mu \\ y \rightarrow v \\ z \rightarrow w \end{matrix} \quad \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} = [\mu] \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix}$$

To specify the relative permeability for an anisotropic material, enter the μ_1, μ_2, and μ_3 values in the **Value** boxes of the **T(1,1)**, **T(2,2)**, and **T(3,3)** rows, respectively. If the relative permeability is the same in all directions, use the same value for μ_1, μ_2, and μ_3. These values can also be entered as variables.

Anisotropic Relative Permittivity Tensors

The relative permittivity tensor for an anisotropic material is described by

$$\varepsilon = \begin{bmatrix} \varepsilon_1\varepsilon_0 & 0 & 0 \\ 0 & \varepsilon_2\varepsilon_0 & 0 \\ 0 & 0 & \varepsilon_3\varepsilon_0 \end{bmatrix}$$

where

- ε_1 is the relative permittivity of the material along one tensor axis.
- ε_2 is the relative permittivity along the second axis.
- ε_3 is the relative permittivity along the third axis.
- ε_0 is the permittivity of free space.

The relationship between *E* and *D* is then

$$\begin{matrix} x \rightarrow \mu \\ y \rightarrow v \\ z \rightarrow w \end{matrix} \quad \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} = [\varepsilon] \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$

To specify the relative permittivity for an anisotropic material, enter the ε_1, ε_2, and ε_3 values in the **Value** boxes of the **T(1,1)**, **T(2,2)**, and **T(3,3)** rows, respectively. These values can also be entered as variables.

Anisotropic Conductivity Tensors

The conductivity tensor for an anisotropic material is described by

$$[\sigma] = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}$$

where

- σ_1 is the relative conductivity along one axis of the material's conductivity tensor.
- σ_2 is the relative conductivity along the second axis.
- σ_3 is the relative conductivity along the third axis.

The relationship between *J* and *E* is then:

$$\begin{matrix} x \rightarrow \mu \\ y \rightarrow v \\ z \rightarrow w \end{matrix} \quad \begin{bmatrix} J_x \\ J_y \\ J_z \end{bmatrix} = [\sigma] \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$

To specify the conductivity for an anisotropic material, enter the σ_1, σ_2, and σ_3 values in the **Value** boxes of the **T(1,1)**, **T(2,2)**, and **T(3,3)** rows, respectively. The values of σ_1 and σ_2 apply to axes that lie in the xy cross-section being modeled. The values of σ_3 apply to the z-component. These values affect current flowing in dielectrics between the conductors. These values can also be entered as variables.

Anisotropic Dielectric Loss Tangent Tensors

The dielectric loss tangent tensor for an anisotropic material is described by

$$[\tilde{\varepsilon}] = \begin{bmatrix} \varepsilon'_1(1 - j\tan\delta_1) & 0 & 0 \\ 0 & \varepsilon'_2(1 - j\tan\delta_2) & 0 \\ 0 & 0 & \varepsilon'_3(1 - j\tan\delta_3) \end{bmatrix}$$

where

- $\tan\delta_1$ is the ratio of the imaginary relative permittivity to the real relative permittivity in one direction.

$$\tan\delta_1 = \frac{\varepsilon''_1}{\varepsilon'_1}$$

- $\tan\delta_2$ is the ratio of the imaginary relative permittivity to the real relative permittivity in the second direction.

$$\tan\delta_2 = \frac{\varepsilon''_2}{\varepsilon'_2}$$

- $\tan\delta_3$ is the ratio of the imaginary relative permittivity to the real relative permittivity in the third orthogonal direction.

$$\tan\delta_3 = \frac{\varepsilon''_3}{\varepsilon'_3}$$

ε'_1

- ε'_2 ε'_3

, , and are the real relative permittivities specified earlier.ε

The relationship between *D* and *E* will then be

$$\begin{matrix} x \rightarrow \mu \\ y \rightarrow \nu \\ z \rightarrow w \end{matrix} \quad \begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} = [\bar{\varepsilon}] \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix}$$

To specify the electric loss tangent for an anisotropic material, enter the $\tan\delta_1$, $\tan\delta_2$, and $\tan\delta_3$ values in the **Value** boxes of the **T(1,1)**, **T(2,2)**, and **T(3,3)** rows, respectively. These values can also be entered as variables.

Anisotropic Magnetic Loss Tangent Tensors

The magnetic loss tangent tensor for an anisotropic material is described by

$$[\bar{\mu}] = \begin{bmatrix} \mu'_1(1 - j\tan\delta_{M1}) & 0 & 0 \\ 0 & \mu'_2(1 - j\tan\delta_{M2}) & 0 \\ 0 & 0 & \mu'_3(1 - j\tan\delta_{M3}) \end{bmatrix}$$

where

- $\tan\delta_{M1}$ is the ratio of the imaginary relative permeability to the real relative permeability in one direction.

$$\tan\delta_{M1} = \frac{\mu''_1}{\mu'_1}$$

- $\tan\delta_{M2}$ is the ratio of the imaginary relative permeability to the real relative permeability in the second direction.

$$\tan\delta_{M2} = \frac{\mu''_2}{\mu'_2}$$

- $\tan\delta_{M3}$ is the ratio of the imaginary relative permeability to the real relative permeability in the third orthogonal direction.

$$\tan\delta_{M3} = \frac{\mu''_3}{\mu'_3}$$

μ'_1

- μ'_2 μ'_3

, , and are the real relative permeabilities specified earlier.

The relationship between *B* and *H* will then be

$$\begin{matrix} x \rightarrow \mu \\ y \rightarrow v \\ z \rightarrow w \end{matrix} \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix} = [\bar{\mu}] \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix}$$

To specify the magnetic loss tangent for an anisotropic material, enter the $\tan\delta_{M1}$, $\tan\delta_{M2}$, and $\tan\delta_{M3}$ values in the **Value** boxes of the **T(1,1)**, **T(2,2)**, and **T(3,3)** rows, respectively. These values can also be entered as functions.

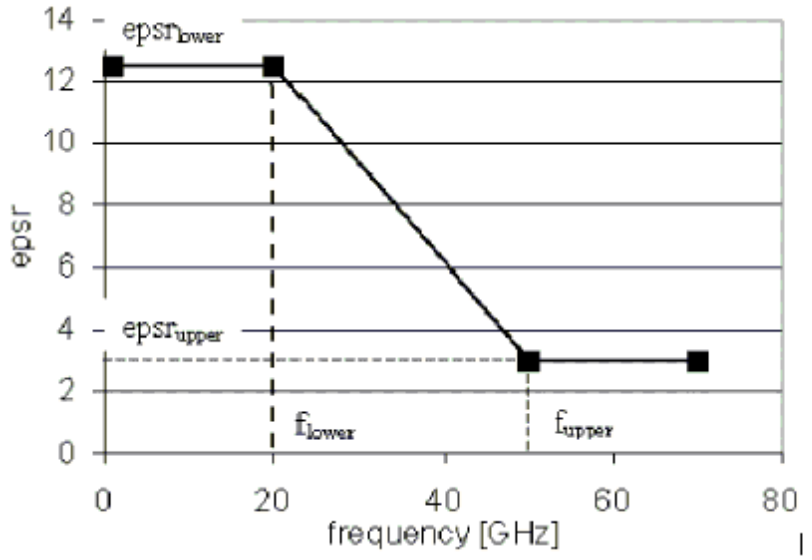
Anisotropic Materials and Ports

An anisotropic material can be in contact with a port under the following conditions:

- One principal axis of the anisotropic material is aligned normal to the port.

Frequency-Dependent Material Properties

The properties of some materials vary with the frequency of the field excitation. This frequency dependence is often linear within a certain frequency range and constant outside of the frequency range, as shown below,



where

- $epsr_{lower}$ is the relative permittivity of a material below the frequency range 20 - 50 GHz.
- $epsr_{upper}$ is the relative permittivity of a material above the frequency range 20 - 50 GHz.
- f_{lower} is the lower frequency, below which the material property is constant.
- f_{upper} is the upper frequency, above which the material property is constant.

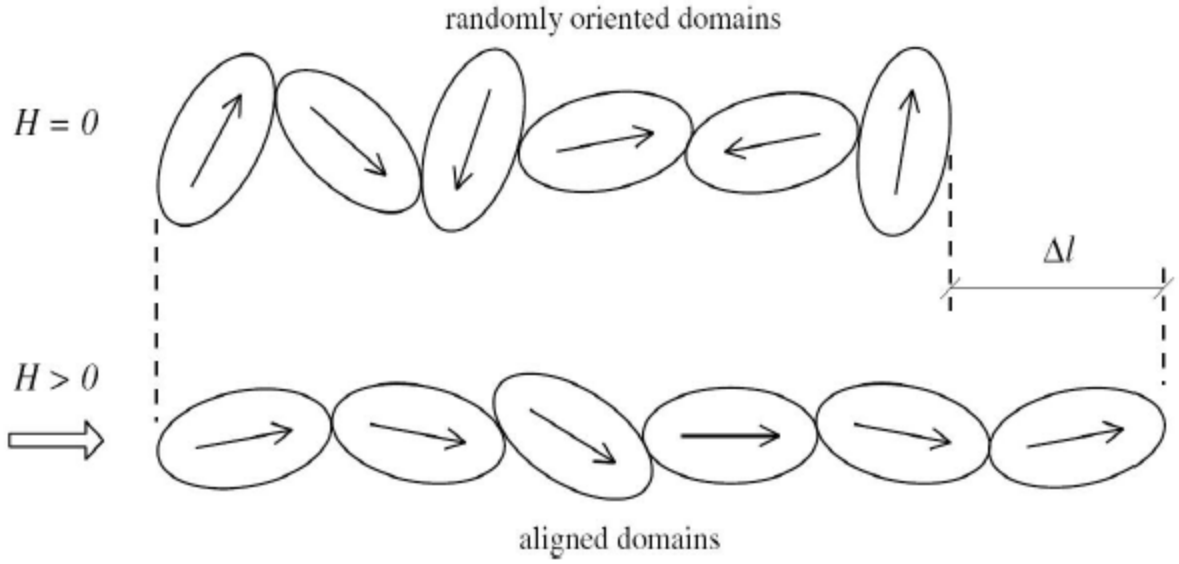
In general, to account for this variance within a given frequency range, use the **Piecewise Linear Material Input** window dialog box to specify a property's values at frequencies below and above the frequency range. Based on these values, Maxwell automatically creates a linear dataset that specifies the property's values at the desired frequencies during solution generation. This dataset can be modified with additional points if desired.

> **Note:** The 3D AC conduction solver only solves the function inserted in the material property field and does not check the upper and lower limits.

Magnetostriction Modeling of Magnetic Material

Magnetostriction was first observed by J. Joule in 1842. Currently it is typically described as "deformation of a body" as a result of its magnetization. The elongation or contraction in the direction of applied magnetic field is usually between 10^{-5} and 10^{-3} and accompanied by the opposite sign changing in the transverse direction, so that the volume remains almost the same. The phenomenon of magnetostriction can be regarded as an energy transduction (or transformation) from mechanical to magnetic and vice-versa. It can be described as a bidirectional magneto-mechanical coupling between the mechanical and magnetic fields in the magnetostrictive material. The origin of this phenomenon can be traced to the alignment of the

magnetic domains inside the material. The following figure is a schematic of the effect simplified to one dimension. When no magnetic field is applied to the system, the series of domains have randomly oriented magnetic moments. Whereas when a magnetic field is applied, the series of domains rotate to partially align themselves to the magnetic field direction resulting in a change in length Δl. Applying a sufficiently high magnetic field will result in perfectly aligned domains, in which case the material achieves the maximum magnetically induced strain (i.e. peak magnetostriction). Magnetostriction requires the magnetic domains to be longer in one dimension than the other two to obtain a change in length when the domain rotates. This, in general, results in anisotropy of the crystal structure of the material.



Magnetostriction is considered to be one of the main sources of noise in Transformers. When an alternating voltage is applied to one or more windings of a transformer, a magnetic flux is generated in the transformer core laminations made of grain oriented electrical steel. The nonlinear anisotropic property of magnetostriction implying alternating changes of the core dimensions due to the varying magnetic flux in the laminations. Those magnetostrictive forces cause core vibrations which are transmitted to the tank via the insulation oil and the core clamping points. Part of the mechanical energy is eventually radiated by the tank walls as noise. In electric motors, magnetostrictive forces also contribute to motor vibration and noise.

On the other hand, magnetostrictive materials convert energy between the mechanical and magnetic domains (inverse magnetostriction). They deform in response to applied magnetic fields and change their magnetic state when stressed. Some materials (such as Terfenol-D, Galfenol) have the ability to produce large magnetostrictive strains at moderate fields. Short response times (in the millisecond range) combined with resolutions in the order of microstrains make these materials well-suited to precision sensing and actuation mechanisms.

Due to the complexity of the mechanism of magnetostriction, it is very difficult to predict it with analytic models. In general, empirical methods based on statistics and dimensional basic parameters are used by most transformer manufacturers and sensor producers. Those

approaches present limitations when applied to new designs and do not enable accurate parametric studies. Therefore, prediction models based on finite element formulations will be utilized to describe accurately the complex interactions of the various design parameters and the coupling of the physical fields.

The finite element method (FEM) is widely used in engineering practice because using the irregular grids it can model complex inhomogeneous and anisotropic materials and represent complicated geometry. The FEM discretization produces a set of matrix differential equations. Because of the nonlinearity, the matrices generally are dependent of the solution vectors, so an iteration method such as Newton-Raphson method should be used to solve these nonlinear matrix equations. Namely, the nonlinear matrix equations are linearized for each nonlinear iteration. The linearized matrix equations may be solved by either a direct or iterative matrix solver.

The governing equation of magnetic field is given by:

$$\sigma \frac{\partial A}{\partial t} + \nabla \times \left(\frac{\nabla \times A}{\mu} \right) = J$$

(equation 1)

Where **A** is the magnetic vector potential, **σ** is the electrical conductivity, and **μ** is the magnetic permeability of material. **J** represents the current density.

The magnetic flux density **B** can be expressed as:

$$B = \nabla \times A$$

(equation 2)

The relationship between **B** and **H** (magnetic field) is based on the constitutive law:

$$B = \mu H$$

(equation 3)

For a discretized mechanical system, the governing equilibrium equation can be written as

$$Kr + R_0 = R$$

(equation 4)

Where K is the system stiffness matrix, is the body force vector, and R the vector of external force, r is vector of node displacement. The strain-displacement relationship can be written as

$$S = \nabla r$$

(equation 5)

Where **S** is the strain tensor. And the stress-strain relationship can be described by

$T=CS$ (equation 6)

where **T** is the stress tensor, and **C** is stiffness matrix.

When a magnetic field is applied to a magnetostrictive material, strain occurs within the material placed in a magnetic field in response to the flux density within the material, while the stress gives feedback to affect the magnetic field. For coupled method, we have:

$$\begin{bmatrix} B \\ S \end{bmatrix} = \begin{bmatrix} \mu & d \\ d^T & C^{-1} \end{bmatrix} \begin{bmatrix} H \\ T \end{bmatrix}$$

(equation 7)

$$d^T = \left. \frac{\partial \lambda}{\partial H} \right|_T$$

(equation 8)

$$d = \left. \frac{\partial B}{\partial T} \right|_H$$

(equation 9)

Where **d** and $\mathbf{d^t}$ are magneto-mechanical coupling coefficients, **λ** is magnetostriction induced by magnetic field **H**. The magnetic field and mechanical stress field are solved together, which means both fields are solved on the same mesh simultaneously by using the one matrix solver.

For decoupled method, the magnetostriction may be determined once the magnetic field distribution is obtained from FEM simulation. Hence an elastic analysis is reduced to the problem of finding the stress due to an external load caused by a force applied to a material with a previously known magnetostriction, the latter unrelated to the stress. Magnetostriction is transformed into stress with the usage of Hooke's law. Equation (7) can be rewritten as:

$$B = \mu H + dT$$

(equation 10)

$$S = C^{-1}T + d^T H$$

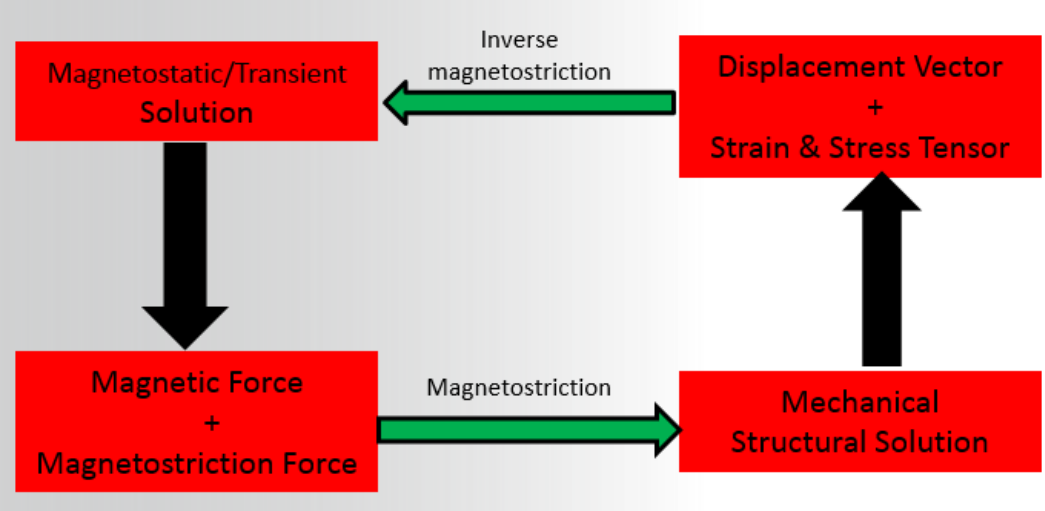
(equation 11)

where

$d^T H$: magnetostrictive, deformation induced by magnetic field **H**.

dT: inverse magnetostriction, impact of stress **T** on magnetic field **B**.

By using the decoupled method, the magnetic field and stress field can be solved independently with different mesh and different solver, and magnetostriction and inverse magnetostriction can be involved in either one or both. Compared to the coupled method, the decoupled method is more flexible and general. The process for the decoupled method is shown in the following figure:



30 - Maxwell Examples

Maxwell provides numerous examples for customer use.

- Maxwell 2D and 3D examples are stored in the **Examples\Maxwell** sub-directory within the **AnsysEM** installation directory.
- Rmxprt examples are stored in the **Examples\RMxprt** sub-directory within the **AnsysEM** installation directory.

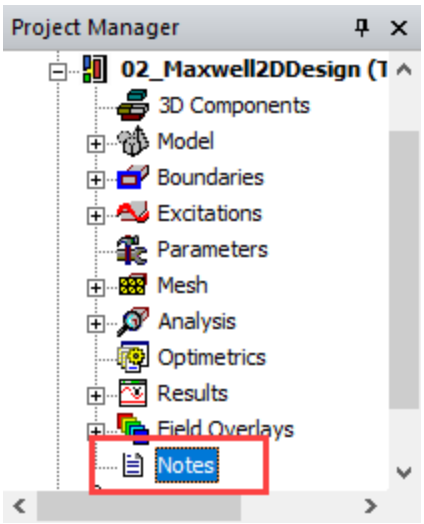
Related Topics

Maxwell 2D and 3D Examples

RMxprt Examples

Maxwell 2D and 3D Examples

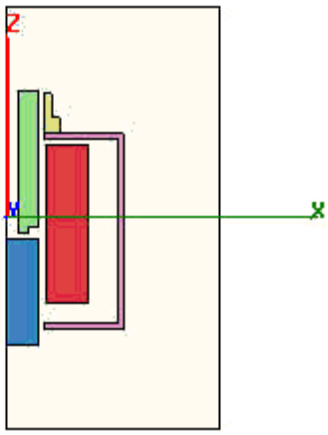
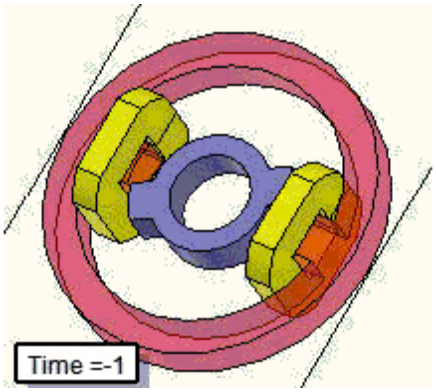
Maxwell 2D and 3D example designs are stored in the **Examples\Maxwell** sub-directory within the **AnsysEM** installation directory. Detailed information for each design is included in the **Notes** in the design's project tree.

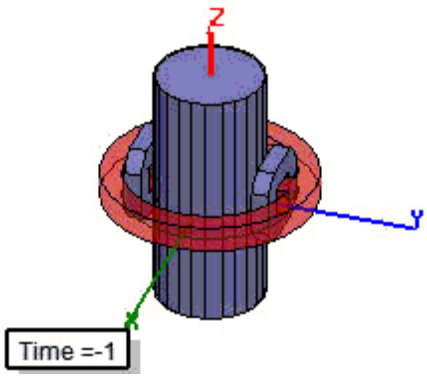
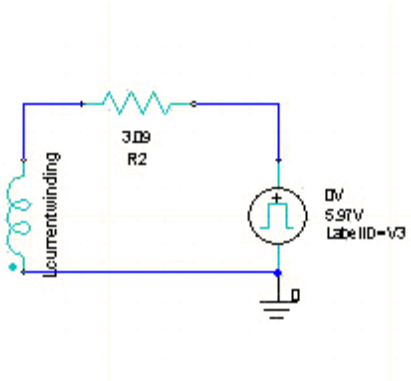


> **Note:** The **Ex_MS_4_ECE_Linear_Movement.aedt** and **WirelessCharging.aedt** designs cannot be used on Linux platforms because they contain **TwinBuilder** designs. **TwinBuilder** is not supported on Linux.

The designs below, found in the **Examples\Maxwell\Actuators** sub-directory, are those modeled in the three Maxwell *Getting Started Guides*.

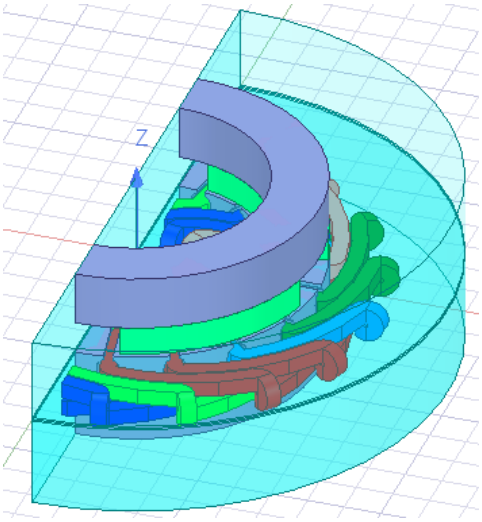
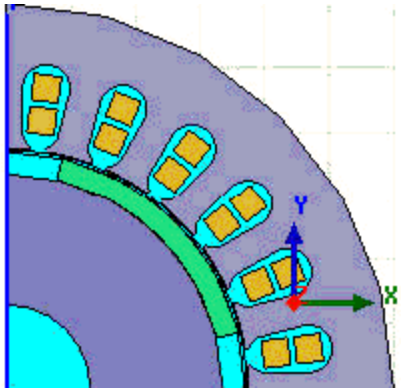
Examples\Maxwell\Actuators	The solenoid model shown is a 2D axisymmetric

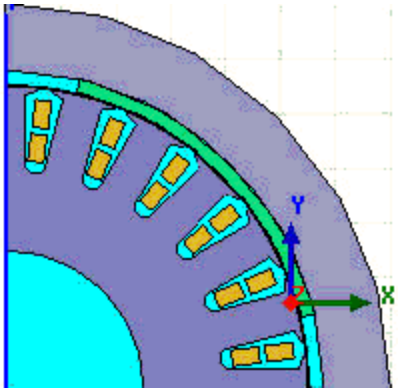
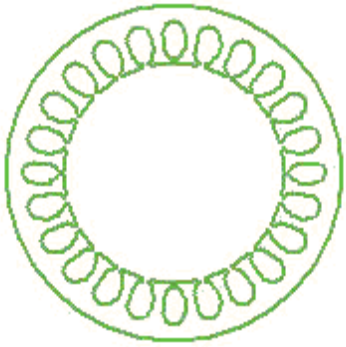
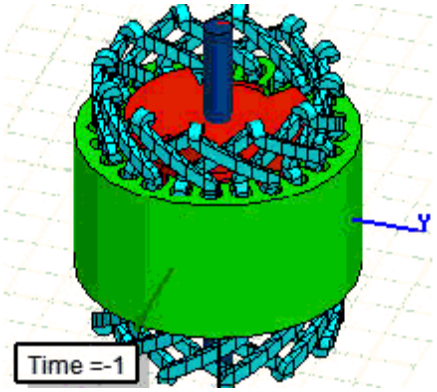
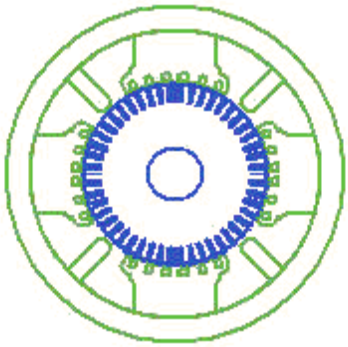
Solenoid.aedt 	simulation of a 3D actuator that demonstrates rotational symmetry about a central axis. The problem is a magnetostatic model that uses parametric analysis to determine the force on the core as various locations along the length of the stroke. This is a non-linear problem incorporating both non-linear steel and a permanent magnet. This problem is used in *Getting Started with Maxwell: A 2D Magnetostatic Solenoid Problem*.
Examples\Maxwell\Actuators\ Rotational_actuator.aedt 	The rotational actuator is another magnetostatic example usable for the parametric study of the torque and inductance characteristics vs. position. Possible variation (s) of the setup can include using ferromagnetic materials with different magnetic (non-linear) characteristics and the respective impact on torque and inductance characteristics. This model is used in the *Getting Started with Maxwell: Designing a Rotational Actuator* guide.
Examples\Maxwell\Actuators\ Rotational_Act_TR.aedt	This model is a modified version of the Rotational Actuator model listed above. It is configured with a motionband for rotating the inner armature. It is a transient problem using an external circuit to drive the coils with a pulse voltage. The results demonstrate the torque, position, and current versus time as the inner armature object moves with time. This model is used in the *Getting Started with Maxwell: Transient Problem* guide.

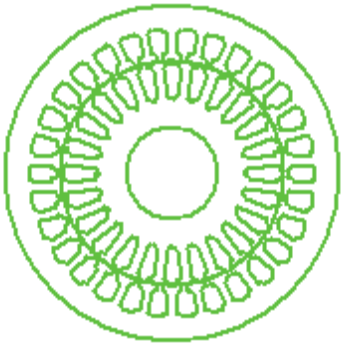
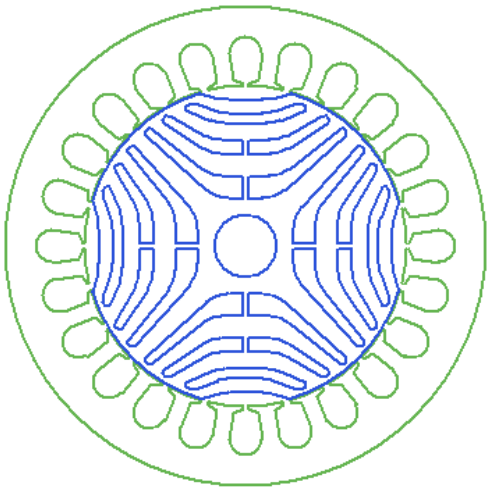
	
Examples\Maxwell\Actuators\ ExternalCircuit.aedt 	This project is the Maxwell Circuit Editor model used in the transient rotational actuator problem to drive the coils with a pulse input.

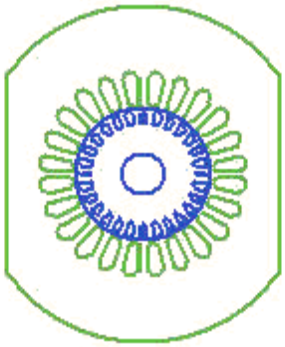
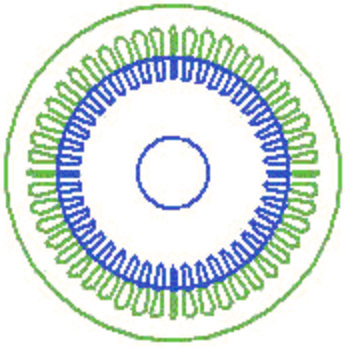
RMxprt Examples:

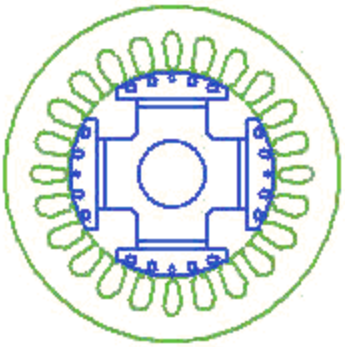
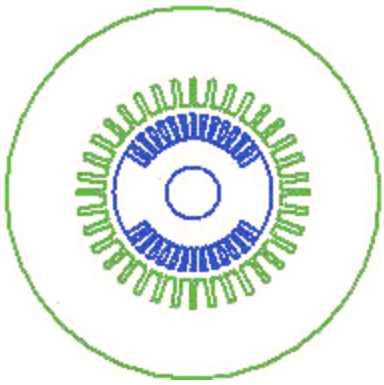
\RMxprt\afpm **afassm_1.aedt** **afassm_2.aedt** **afbldc_1.aedt** **afbldc_2.aedt** **afbldc_3.aedt** **afbldc_4.aedt**	Axial Flux, Permanent Magnet Generic Rotating Machines • 3D representation of **afbldc_2.aedt** is shown

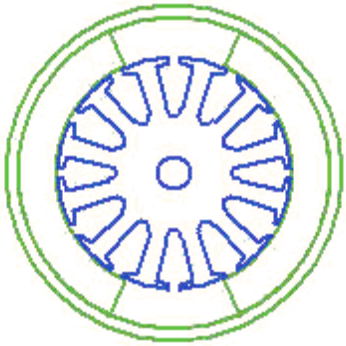
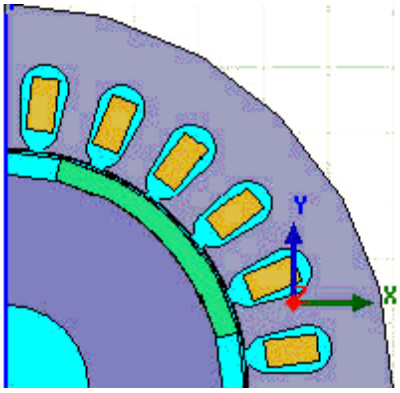
	
\RMxprt\assm **assm-1.aedt** **assm-2.aedt** **assm-3.aedt** **assm-4.aedt** **assm-5.aedt** **assm-6.aedt** 	Adjustable Speed Synchronous Machine • ASSM can be used in Motor Mode or Generator Mode, this example is in motor mode
\RMxprt\bldc **QuickStart.aedt** **ws-1.aedt** **ws-2.aedt** **ws-3.aedt** **ws-4.aedt** **ws-5.aedt**	Brushless Permanent Magnet DC Motor • ws-1 to ws-5 are examples of rotor PM Pole Type 1-5 as defined the "Rotor" section • ws-6 is an example of an outside rotor as shown

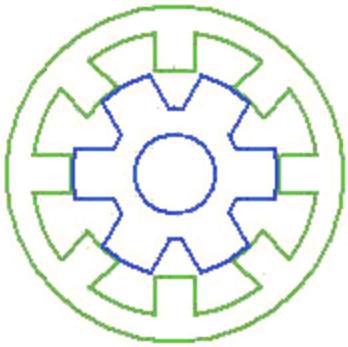
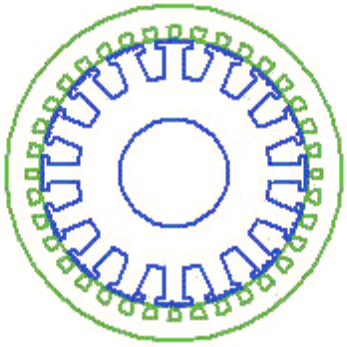
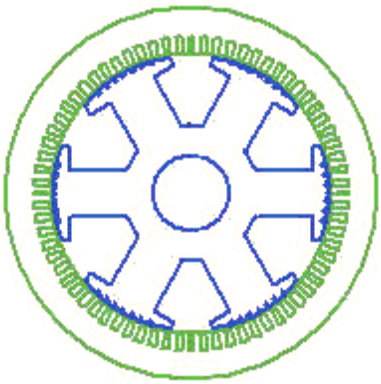
ws-6.aedt	
\RMxprt\cpsg\cpsg_1.aedt 	Claw Pole Synchronous Machine 3D Representation is shown. 
\RMxprt\dcm **z_500.aedt** **zf_500.aedt** 	DC Machines • these do not have any permanent magnets
\RMxprt\dfig	Generic Rotating Machines

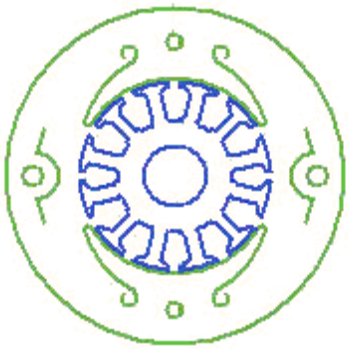
dfig_1.aedt **dfig_2.aedt** 	**dfig_2.aedt** is shown.
\RMxprt\grm **AxialCageRotor.aedt** **CageRotor.aedt** **SalientPole.aedt** **StepMotor.aedt** **SynRRotor.aedt**	Generic Rotating Machines • SynRRotor is shown 
\RMxprt\hysm **ipm_HysRotor.aedt** **ipm_SolidRotor.aedt**	Generic Rotating Machines
\RMxprt\indm1\3hp.aedt	Single Phase Induction Motor

	
\RMxprt\indm3 **tsyde-22.aedt** **ylew-95.aedt** **yz200-6.aedt** **yz200-24.aedt** **yzd132-4.aedt** **yzd132-4G.aedt** **yzd132-8.aedt**	Three Phase Induction Motor • These are 22, 95, 6, 24, 4, and 8 pole motor variations • 8 pole machine is shown. 
\RMxprt\ipm **ipm_1.aedt** **ipm_2.aedt** **ipm_3.aedt** **ipm_4.aedt** **ipm_5.aedt** **ipm_6.aedt**	Generic Rotating Machine (Interior Permanent Magnet)
\RMxprt\lssm **sm-1.aedt** **sm-2.aedt** **sm-3.aedt** **sm-4.aedt** **sm-5.aedt** **sm-6.aedt**	Line Start Permanent Magnet Synchronous Motor • sm-1 to sm-8 represent the 8 rotor pole types available • sm-8 is shown.

sm-7.aedt **sm-8.aedt**	
\RMxprt\manual **ASSM_4p50Hz550W.aedt** **AssymetricWinding.aedt** **BLDC_4p1500rpm550W.aedt** **CPSG_4p50Hz550VA.aedt** **DCM_4p1100rpm500kW.aedt** **DCM_4p3000rpm500kW.aedt** **IndM1_2p60Hz3hp.aedt** **IndM3_6p50Hz11kW.aedt** **LSSM_4p50Hz550W.aedt** **NSSM_2p3000rpm125MVA.aedt** **PMDC_2p2400rpm75W.aedt** **PMSG_4p50Hz550W.aedt** **QuickStart.aedt** **SRM_8-6p1500rpm550W.aedt** **SynM3_6p50Hz538kW.aedt** **UniM_2p12500rpm100W.aedt**	
\RMxprt\nssm **nssm-1.aedt** 	

\RMxprt\pmdc\ stndj-1.aedt 	Permanent Magnet DC Motor • These are brush type motors
\RMxprt\pmsg\ pmsg-1.aedt 	Adjustable Speed Synchronous Machine • ASSM can be used in Motor Mode or Generator Mode, this example is a generator
\RMxprt\srm\srm-1.aedt	Switched Reluctance Motor

	
\RMxprt\syng3 **bjs7-6-1.aedt** **bjs10-61.aedt** **bsm.aedt** **g2-5.aedt**	Three Phase Synchronous Machine • This can be either motor or generator mode, these examples are generators • Model g2-5 is shown 
\RMxprt\synm3 **Bjs7_mot.aedt** 	Three Phase Synchronous Machine • This can be either motor or generator mode, this example is a motor

\RMxprt\unim\ac120_i.aedt 	Universal Motor Example

31 - Maxwell Terminology

This section defines the terminology used in the Ansys Maxwell® help topics. Terms are listed in alphabetical order.

Glossary: A

advanced nonlinear options

For the magnetostatic solution type, the advanced nonlinear options allow you to perform more sophisticated simulations of the magnetostatic field. (Nonlinear materials are materials in which the magnetic flux density varies nonlinearly with the magnetic field.)

anisotropic materials

Anisotropic materials exhibit different values of the respective material property in different directions of the three-orthogonal coordinate system used to define it. Any of the material properties can be anisotropic: electric conductivity, dielectric permittivity, magnetic permeability. Mathematically, tensors are used to express the respective anisotropic properties.

Glossary: B

B-H curve

In nonlinear materials, when a material has a permeability that varies with the flux density, a B-H curve is used to describe the material's nonlinear behavior. The B refers to the magnetic flux density. The H refers to the magnetic field intensity.

Glossary: C

capacitance

Capacitance is a quantity used to describe the ability of a pair of corresponding electrodes to store electric energy. In an equivalent formulation, the storage of energy corresponds to charge accumulation on the respective armatures while a voltage is established between them. Mathematically, capacitance is the positive defined ratio between the charge accumulated on one of the armatures and the voltage between the two armatures.

cascaded windows

Cascaded windows are a set of partially overlapping windows stacked on top of each other.

coil terminals

A coil terminal is used to apply excitations to coils. It can be a 2D sheet object obtained, for example via a section command (for closed coils that are totally contained within the region where the field problem solution is being computed), or it can be materialized by the intersection between the problem region and open-ended conductors. It is assumed that the current flow is perpendicular to the coil terminal.

conduction path

The conduction path is a sequence of conductors (objects with non-zero conductivity) through which current flows.

conductivity

Conductivity (electric) represents the ability of a material to allow electric current to flow through it.

conductor

Conductor are, in general, objects with non-zero value of conductivity. The way such objects are treated by the software depends upon the user-defined conductivity threshold value.

cost function

In an optimization setup, a cost function is calculated based on goal values specified for one or more solution quantities. The chosen Optimetrics engine automatically changes the design parameter values to fulfill the objective of the optimization. The cost function can be based on any quantity that can be computed directly by Maxwell or by the field calculator.

current density terminals

This refers to any terminal used to specify the current density of an object.

cutplane

A cutplane is any plane that cuts through the model. Cut planes can cut through the model at any angle and may be moved to a new position to observe a difference in plotted fields.

Glossary: D

damping (linear motion)

A value for damping for linear motion can be calculated by the windage and friction loss in Watts divided by the speed in m/sec (at which those losses are defined); this is Damping with resulting units of (W / (m/sec)^2) or (N*m * sec/m).

damping (rotational motion)

A value for damping for rotational motion (for electric machines) can be calculated by the windage and friction loss in Watts divided by the speed in rad/sec (at which those losses are defined); this is Damping with resulting units of (W / (rad/sec)^2) or (N*m * sec/rad).

design variation

A single combination of variable values that is solved during a parametric or optimization setup.

dielectric loss tangent

The dielectric loss tangent is a material property characteristic for lossy dielectrics, i.e. dielectrics which exhibit power losses.

Dirichlet boundary

The Dirichlet boundary (or value boundary) allows you to prescribe the value of the function (solution) on the respective surface of the solution domain where it is applied. Alone or together with other applied boundary conditions, it ensures the uniqueness of the field solution calculated by Maxwell.

Glossary: E

eddy current

Eddy currents are the currents due to the electric field induced by the time varying magnetic field in conductors.

eddy effect

The eddy effect refers to the current flow induced by the time-varying magnetic field in conductors. It can be turned on or off depending on the characteristics of the application. Turning the eddy effect off in conductors (when their effects are estimated to be negligible) can save significant computing resources.

electric coercivity

This refers to the electric field that exists when the electric flux is zero.

electric retentivity

The electric retentivity refers to the value of the electric flux density when the electric field is zero.

electric flux

Electric flux is defined as the integral of the electric flux density vector over a specified surface. According to Gauss's law, the electric flux calculated for a closed surface is equal to the total electric charge enclosed by the respective surface.

electric susceptibility

This material property refers to a material's ability to become polarized.

executive parameters

Executive parameters are parameters of great practical significance that are calculated directly by the Maxwell solvers for all specified objects.

Glossary: F

finite element analysis

Finite element analysis (FEA) is a numerical method of solving physical domain partial differential equations (such as Maxwell's equations for electromagnetic fields) by which the solution domain is divided into a set of geometric entities such as tetrahedrons.

finite element mesh

The finite element mesh for a Maxwell problem is the assembly of all tetrahedral finite elements used to discretize the solution space where the respective electromagnetic fields are calculated.

floating boundaries

In electrostatics, floating boundaries are similar to Dirichlet boundaries in that the entire surface is at a single potential, except that you need not specify the potential as a known a value. This is used to model conductors at unknown potentials.

floating conductor

Floating conductors are conductors at which the potential is unknown.

Glossary: G

Genetic Algorithm

Optimization method of the experiment tool with automatic parameter variation and target function determination.

goal

In an optimization setup, a goal is the value of a solution quantity that you want to be achieved during the optimization. A goal is represented as one row in the cost function table. Each cost function defined in an optimization setup must include at least one goal.

grid plane

The grid plane is the plane on which you want to display the grid in the active view window. Also called the drawing plane.

Glossary: H

hotkeys

Hotkeys are keystroke combinations that execute more complex commands without the need for accessing the commands in the menu bars or tool bar icons.

Glossary: I

imaginary permeability

This material property refers to the imaginary component of the permeability of a material. See **permeability**.

imaginary anisotropic relative permeability

This material property refers to the imaginary component of the anisotropic relative permeability of an object. See **anisotropic permeability**.

imaginary relative permeability

This material property refers to the imaginary component of the relative permeability of a material. See **relativepermeability**.

impedance

Impedance represents the characteristic of an electric circuit element to impose a certain dependency between the voltage between a pair of terminals and the corresponding AC current flowing between the two terminals.

independent generator

The generator supplies electrical power to a load inductor. For an AC generator, the resistance and inductance of the load inductor are determined by the rated output power, the rated voltage, and the rated power factor. For a DC generator, the resistance of the load resistor is determined by the rated output power and the rated voltage.

inductance

Inductance represents the property of a magnetic circuit to relate its magnetic flux to the current which produced it. It is the expression of the property of magnetic devices to store magnetic energy.

infinite bus

When the generator is connected to an infinite bus of the power system, the terminal voltages remain constant and the output power is determined by the output currents.

insulating boundary

An insulating boundary is a convenient way of preventing the current flow across the boundary. This boundary condition is placed between conductors and allows you to avoid creating small separating spaces between conductors for insulation purposes.

insulator

An insulator refers to a material that does not allow the current to flow through it (typically with zero electric conductivity). The way such objects are treated by the software depends upon the user-defined insulation threshold value.

isotropic materials

Isotropic materials refer to materials with properties independent of the direction (orientation) of the axes of the coordinate system in which they are defined.

Glossary: L

lamination model

Lamination modeling allows you to specify a stacking factor and stacking direction, which represents the direction perpendicular to the plane of the lamination. This lamination model is an alternative way of specifying anisotropic behavior when using laminations.

line impedance

The line impedance of an object is the impedance of a single current loop.

LMB

This is an acronym for the left mouse button. The left mouse button is used to choose commands and select points, surfaces, and objects.

Lorentz force

The Lorentz force is the force on an object due to the currents in the magnetic field, neglecting forces due to the effects of material properties.

Lorentz torque

The Lorentz torque is the torque of an object due to the currents in a magnetic field, neglecting torques due to the effects of material properties.

lumped capacitance

Lumped capacitance is the capacitance between two conductors or groups of conductors.

Glossary: M

magnetic coercivity

This refers to the magnetic field that exists when the magnetic flux is zero.

magnetic flux

The magnetic flux is the total of all the lines of force in a magnetic field.

magnetic loss tangent

The magnetic loss tangent allows you to represent a dielectric that dissipates the power of a high-frequency magnetic field. The smaller the loss tangent, the less lossy the material.

magnetic retentivity

In nonlinear materials, the magnetic retentivity (or remanence) refers to the value of the magnetic flux density when the magnetic field is zero.

magnetic susceptibility

The magnetic susceptibility is a measure of the ability of the material to become magnetized.

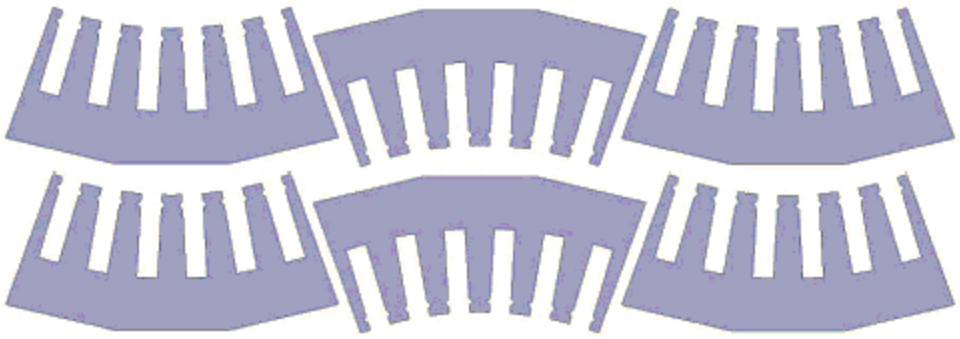
matching boundaries

Matching boundary conditions force the field solution at one surface (the Dependent surface) to match the magnitude and orientation (or opposite orientation) of the field at the Independent surface.

material consumption

Whole-circle laminations – Considers a minimum section of square “raw steel” from which the lamination could be cut. The calculation takes the rotor or stator diameter as the length of the square and calculates the weight based on the material density.

Lamination sectors – For large size machines, a lamination layer of the stator core may be laminated sector-by-sector, instead of a whole circle lamination. The number of Lamination Sectors indicates how many lamination sectors are needed to form a lamination layer. Lamination sectors are punched as shown below so that material consumption is minimized. There is a 3mm allowance in the edge and between two sectors.



material coordinate system

The material coordinate system is the coordinate system used to define the vector or anisotropic material property of objects. Maxwell supports the following three material coordinate systems: Cartesian, cylindrical, and spherical.

mesh refinement

Mesh refinement refers to the process of successively breaking down large error finite elements into smaller ones for the purpose of continuously increasing the accuracy of the finite element calculation. Maxwell uses proprietary adaptive mesh refinement algorithms designed to automatically detect and refine the mesh in regions with the largest field error.

multi-processing

Using more than one processor to solve a Maxwell design. Requires a multiple processor license. Using multiple processors (if physically available on the computer) speeds up the matrix solution process phase. Thus, shorter solution times can be achieved with significant saves in absolute terms, particularly for large models of 100,000 finite elements or more. There are no negative side effects if the number of processors to be used is set to more than are available on the respective computer. In this case, the actual number of available processors will be used in the matrix solution process.

Glossary: N

natural boundary

A natural boundary is a boundary condition at the interface between different objects. You do not need to apply this boundary condition yourself, as Maxwell enforces it internally. The natural boundary condition reflects the particular local form of Maxwell's equations at the interface between objects where there is an sudden change in the material properties. When crossing such a surface of discontinuity, Maxwell assumes the following behavior (in average):

- The normal component of B is continuous.
- The normal component of D has a jump equal with the local superficial charge density (on the discontinuity surface).
- The tangent component of H is continuous if there are no current sheets on the discontinuity surface (has a jump equal with the density of the current sheet if they are present on the discontinuity surface), and so on.

Neumann boundary

For magnetic problems with T-Omega formulation, the homogeneous Neumann boundary condition specifies that the normal component of the H field is zero. For magnetic problems with A-Phi formulation, the homogeneous Neumann boundary condition specifies that the tangential component of the H field is zero. In other words, this boundary condition confines the magnetic field within the space of the problem. If the out-most surfaces of the problem region are not assigned any boundary conditions, they automatically receive a Neumann BC. Because of this, as a rule, the solution space (region size) should be chosen such that the respective region walls are not close to field sources (currents and/or permanent magnets); this allows us to assume the field is a tangent at the region limits.

nodes

Nodes normally refer to the vertices of the tetrahedral finite elements. They can also refer to the connection points of branches in an electric circuit (if external circuits are used to drive coils).

nominal design

The original model on which Optimetrics analyses are based.

nonlinear materials

Nonlinear materials are materials with the constitutive relationship being nonlinear (such as is the case of the dependency between B and H for some magnetic materials).

Glossary: O

outer terminals

Typically, outer terminals are faces of 3D objects that lie on the surface of the problem region.

Glossary: P

parametric analysis

A parametric analysis consists of a sweep of one or more design or project variables for the purpose of analyzing one or more output quantities (Maxwell parameters or quantities calculated by the field calculator).

paste buffer

When you cut or copy an object, such as with the **Edit** commands, the object is stored in the paste buffer. The paste buffer itself is an area of resident memory that is erased and replaced with the newest item that has been cut or copied.

perfect conductor

Perfect conductors are idealized materials with very high conductivity (typically many orders of magnitude above the usual conductivity values). The actual value is influenced by the perfect conductivity threshold value you specify in the solve options.

permanent magnets

Permanent magnets are any materials that generate magnetic flux due to permanent magnetic dipoles in those materials.

permeability

Permeability is a material property characteristic of magnetic materials and relates magnetic flux density to magnetic field strength in objects with no permanent magnetization present.

permittivity

Permittivity is a material property characteristic for dielectrics and relates electric flux density to electric field strength in objects with no permanent electric polarization present.

phasors

Phasors represent a conventional way of representing sinusoidal (AC) quantities such as voltages and currents. In a phasor representation, the amplitude of the signal relates to the length of the rotating vector (phasor) while the initial phase is the starting angle at time t=0. The angular velocity of the phasor rotation relates to the frequency of the signal.

polyline

A polyline is an object composed of one or more straight or curved line segments.

post processing

Post processing is the phase of the analysis when information is extracted from the simulation. Typical results are 2D plots, contour and vector plots of field quantities, results from the field calculator, animations, etc.

press board (pressboard)

An optional material used above and below the lamination stack and as a spacer in manufacturing. In RMxprt, users can specify the press board thickness and whether it is magnetic or non-magnetic.

problem region

The problem region is the region where Maxwell calculates the field solution. This is an all-encompassing object with user-selected material properties and with surfaces that are used to apply the necessary boundary conditions (user-specified or default ones).

Glossary: R

relative permeability

Relative permeability relates the absolute permeability of a magnetic material to the permeability of vacuum ($4\pi 10^{-7}$ H/m).

relative permittivity

Relative permittivity relates the absolute permittivity of a magnetic material to the permittivity of vacuum($4\pi 10^{-7}$ H/m).

resistance

Resistance represents the characteristic of a DC electric circuit element to impose a certain dependency between the voltage between a pair of terminals and the corresponding DC current flowing between the two terminals.

RMB

This is an acronym for the right mouse button.

RMBM

This is an acronym for the right mouse button menu. This menu becomes visible when you hold the right mouse button down while the cursor is in a view window. Depending on the module you are in, you may use this menu to select or deselect objects, or accept or cancel selections or values.

Glossary: S

section

This refers to the **Surfaces/Section** command which creates a cross-section of 3D objects with the XY, YZ, or XZ planes.

seeding

Seeding refers to the process of adding points to your finite element mesh to create a finer one.

self-inductance

Self-inductance of an object or group of objects associates the current flowing in the respective object(s) with the own flux linkage.

skin depth

The skin depth of a material is the distance measured inside a conductor from the surface where the AC field (magnetic/electric) penetrating the conductor is reduced by a factor of 1/e (~ 1/2.71). As general rule is that at 5 skin depths from the surface, the AC fields penetrating inside conductors can be considered to be zero.

$$\delta = \sqrt{\frac{2}{\omega \sigma \mu_0 \mu_r}}$$

slot fill factor

The slot fill factor is the ratio between the cross-sectional area of all conductors in one slot and the entire slot area.

solver

Maxwell has a number of solvers allowing you to choose from electric, magnetostatic, eddy, and transient, depending upon the characteristics of the problem to be solved.

solver residual

The solver residual is an error term that measures the current accuracy of the solution process when using an iterative solver.

superposition

Superposition allows you to combine the results of field solutions, each corresponding to linear problems. Superposition should not be used when problems are nonlinear.

surface

A surface is an object face.

sweep

Sweep can refer to two different topics:

- In the 3D Modeler, sweeping a face or object along a path or vector creates a new 3D object. The object you sweep need not be orthogonal to the path or vector.
- In the Executive Parameters module, a parametric sweep refers a series of variables.

symmetry boundary

Symmetry boundaries model the planes of symmetry in a problem.

Glossary: T

terminal

As a terminal, you can use an exact cross-section of the conduction path, to apply an eligible excitation.

tile window

Existing windows, regardless of their contents, can be tiled either horizontally or vertically.

toolbar

A toolbar contains icons which can be used to shortcut commands.

Glossary: V

value boundary

See **Dirichlet boundary**.

variable sweep definition

A set of variable values within a range that Optimetrics drives Maxwell to solve when a parametric setup is analyzed. A parametric setup can include one or more sweep definitions.

vector functions

Vector functions are used to specify certain vector type properties. For instance, magnetization can be defined using vector properties in a chosen, previously defined coordinate system.

view window

In the 3D Modeler, the view window is the window in which you create your model. The view window gives a 3D view of the model. In this window, you can assign materials to objects, specify boundary conditions, set up any executive parameters, or post process data or fields in the model.

virtual force

Virtual force is the force calculated on an object or group of objects using a special mathematical algorithm that contains the effects of a virtual motion along a generalized coordinate (such as x, y, z). The movement does not actually take place; it is virtual.

virtual torque

Virtual torque is the torque calculated on an object or group of objects using a special mathematical algorithm that contains the effects of a virtual rotation around a coordinate axis (such as x, y, z). The movement does not actually take place; it is virtual.

visibility

Visibility refers to whether or not an object is displayed. You can show or hide an object using the **View>Active View Visibility** commands.

Index

D

N

T

U